

MBS Util Plugin Documentation

Christian Schmitz

May 5, 2025

0.1 Introduction

This is the PDF version of the documentation for the Xojo Plug-in from Monkeybread Software Germany.
Plugin part: MBS Util Plugin

0.2 Content

• 1 List of all topics	3
• 2 List of all classes	177
• 3 List of all modules	181
• 4 List of all global methods	183
• 5 All items in this plugin	195
• 45 List of Questions in the FAQ	1255
• 46 The FAQ	1265

Chapter 1

List of Topics

• 29 Process	631
– 29.1.1 class Application	631
* 29.1.3 ApplicationCreatorCodeMBS as string	631
* 29.1.4 ApplicationFileMBS as folderitem	632
* 29.1.5 ApplicationNameMBS as string	632
* 29.1.6 ArgumentsMBS as String()	632
* 29.1.7 BundleFolderMBS as folderitem	633
* 29.1.8 BundleLocalizedStringMBS(key as string) as string	633
* 29.1.9 BundleLocalizedStringMBS(key as string,fromtable as string) as string	633
* 29.1.10 BundleResourceFolderItemLocalizedMBS(ResourceName as string, ResourceType as string, SubDirectory as string) as folderitem	634
* 29.1.11 BundleResourceFolderMBS as folderitem	634
* 29.1.12 HideMeMBS as boolean	634
* 29.1.13 HideOthersMBS as boolean	635
* 29.1.14 LaunchTimeMBS as Double	635
* 29.1.15 ProcessTimeMBS as Double	635
* 29.1.17 FrontmostMBS as boolean	636

• 5 Atomic	195
– 5.1.1 class AtomicFlagMBS	195
* 5.1.3 Clear	196
* 5.1.4 Operator_Compare(other as AtomicFlagMBS) as Integer	196
* 5.1.5 Operator_Convert as Boolean	196
* 5.1.6 Operator_Convert(value as Boolean)	196
* 5.1.7 Test as Boolean	197
* 5.1.8 TestAndSet as Boolean	197
* 5.1.10 Value as Boolean	197
– 5.2.1 class AtomicIntegerMBS	198
* 5.2.3 Add(Value as Int64) as Int64	198
* 5.2.4 BitwiseAnd(Value as Int64) as Int64	199
* 5.2.5 BitwiseOr(Value as Int64) as Int64	199
* 5.2.6 BitwiseXOr(Value as Int64) as Int64	199
* 5.2.7 CompareExchange(byref expected as Int64, desired as Int64) as Boolean	200
* 5.2.8 Decrement as Int64	200
* 5.2.9 Exchange(Value as Int64) as Int64	200
* 5.2.10 Increment as Int64	201
* 5.2.11 Operator_Add(other as Int64) as Int64	201
* 5.2.12 Operator_Compare(other as AtomicIntegerMBS) as Integer	202
* 5.2.13 Operator_Convert as Int64	202
* 5.2.14 Operator_Convert(value as Int64)	203
* 5.2.15 Operator_Subtract(other as Int64) as Int64	203
* 5.2.16 Subtract(Value as Int64) as Int64	203
* 5.2.18 Value as Int64	203
– 5.3.1 class AtomicQueueMBS	204
* 5.3.3 Add(values as AtomicQueueMBS)	204
* 5.3.4 Add(values() as Variant)	205
* 5.3.5 Clear	205
* 5.3.6 Insert(value as Variant)	206
* 5.3.7 Operator_Add(value as Variant) as AtomicQueueMBS	206
* 5.3.8 Operator_Compare(other as AtomicQueueMBS) as Integer	206
* 5.3.9 Operator_Convert as Variant()	207
* 5.3.10 PopFirst as Variant	207
* 5.3.11 PopLast as Variant	208
* 5.3.12 Push(value as Variant)	208
* 5.3.13 Values as Variant()	208
* 5.3.15 Count as Integer	208
* 5.3.16 TopValue as Variant	209

• 15 Declare	299
– 15.1.1 class BlockMBS	299
* 15.1.3 Close	299
* 15.1.4 GetBlockB(tag as Variant = nil) as Integer	300
* 15.1.5 GetBlockBI(tag as Variant = nil) as Integer	300
* 15.1.6 GetBlockBII(tag as Variant = nil) as Integer	300
* 15.1.7 GetBlockBIII(tag as Variant = nil) as Integer	300
* 15.1.8 GetBlockBIIII(tag as Variant = nil) as Integer	301
* 15.1.9 GetBlockV(tag as Variant = nil) as Integer	301
* 15.1.10 GetBlockVI(tag as Variant = nil) as Integer	301
* 15.1.11 GetBlockVII(tag as Variant = nil) as Integer	301
* 15.1.12 GetBlockVIII(tag as Variant = nil) as Integer	302
* 15.1.13 GetBlockVIIIII(tag as Variant = nil) as Integer	302
* 15.1.15 AsyncBoolResult as Boolean	302
* 15.1.16 Synchronous as Boolean	302
* 15.1.18 BlockB(Async_ as boolean, tag as Variant) as boolean	303
* 15.1.19 BlockBI(Async_ as boolean, tag as Variant, value as Integer) as boolean	303
* 15.1.20 BlockBII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer) as boolean	303
* 15.1.21 BlockBIII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer) as boolean	303
* 15.1.22 BlockBIIII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer, value4 as Integer) as boolean	304
* 15.1.23 BlockV(Async_ as boolean, tag as Variant)	304
* 15.1.24 BlockVI(Async_ as boolean, tag as Variant, value as Integer)	304
* 15.1.25 BlockVII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer)	305
* 15.1.26 BlockVIII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer)	305
* 15.1.27 BlockVIIIII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer, value4 as Integer)	305

• 18 Encryption and Hash	345
– ?? Globals	??
* 18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32	353
* 18.1.3 CRC16MBS(data as string) as UInt16	346
* 18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	346
* 18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32	347
* 18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32	347
* 18.1.7 CRC_32OfStrMBS(s as String) as UInt32	347
* 18.1.8 CRC_CCITInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	348
* 18.1.9 CRC_CCITInMemMBS(address as Ptr, length as Integer) as UInt32	348
* 18.1.10 CRC_CCITTOfStrContMBS(s as String, prevCRC as UInt32) as UInt32	349
* 18.1.11 CRC_CCITTOfStrMBS(s as String) as UInt32	349
* 18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String	349
* 18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String	349
* 18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64	349
* 18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64	352
* 18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64	350
* 18.1.21 CurrentUnixTimeMBS as UInt64	353
* 18.1.22 DecodeFromBase32MBS(data as string) as String	353
* 18.1.23 EncodeToBase32MBS(data as string) as String	354
* 18.1.16 GetHash32MBS(s as string) as UInt32	350
* 18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16	351
* 18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string	351

• 29 Process	631
– 29.2.1 module CallDelegateCrashSafeMBS	637
* 29.2.3 CallDelegateCrashSafeMBS(m as DelegateCrashSafeMBS)	638
* 29.2.4 PrintBacktraceMBS	639
* 29.2.6 CallCounter as Integer	639
* 29.2.7 CrashCounter as Integer	639
* 29.2.9 DelegateCrashSafeMBS()	639
– 29.3.1 module CallDelegatesMBS	640
* 29.3.3 CallDelegateOnMainThreadMBS(m as _delegateMBS)	640
* 29.3.4 CallDelegateOnPreemptiveThreadMBS(m as _delegateMBS) as Boolean	640
* 29.3.5 CallDelegateOnPreemptiveThreadMBS(m as _delegatePtrMBS, data as Ptr) as Boolean	641
– ?? Globals	??
* 29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean	643
* 29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean	643
* 29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean	644
* 29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean	645
* 29.4.5 CallMethodMBS(target as object, name as string) as boolean	646
* 29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean	646
* 29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean	647
* 29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean	647
* 29.4.9 CallMethodOnMainThreadMBS(target as object, name as string) as boolean	648
* 29.4.10 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant) as boolean	649
* 29.4.11 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean	649
* 29.4.12 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean	650
* 29.4.14 CountProcessesMBS as Integer	651
* 29.4.13 SetThreadNameMBS(name as string)	651

• 26 Math	535
– ?? Globals	??
* 26.1.3 ACosHMBS(x as Double) as Double	536
* 26.1.4 ACosMBS(x as Double) as Double	536
* 26.1.5 ArithmeticShiftMBS(value as UInt64, count as Integer) as UInt64	537
* 26.1.6 ASinHMBS(x as Double) as Double	537
* 26.1.7 ASinMBS(x as Double) as Double	537
* 26.1.8 ATan2MBS(x as Double, y as Double) as Double	538
* 26.1.9 ATanHMBS(x as Double) as Double	538
* 26.1.10 ATanMBS(x as Double) as Double	539
* 26.1.11 BitClearMBS(value as UInt64, mask as UInt64) as UInt64	539
* 26.1.12 BitCountMBS(value as UInt64) as Integer	539
* 26.1.13 BitExclMBS(value as UInt64, bitNumber as Integer) as UInt64	540
* 26.1.14 BitInclMBS(value as UInt64, bitNumber as Integer) as UInt64	540
* 26.1.15 BitIsSetMBS(value as UInt64, bitNumber as Integer) as Boolean	540
* 26.1.16 BitValMBS(bitNumber as Integer) as UInt64	541
* 26.1.17 BitwiseDiffMBS(x as UInt64, y as UInt64) as UInt64	541
* 26.1.18 BitwiseNAndMBS(x as UInt64, y as UInt64) as UInt64	541
* 26.1.19 BitwiseNOrMBS(x as UInt64, y as UInt64) as UInt64	542
* 26.1.20 BitwiseNotMBS(value as UInt64) as UInt64	542
* 26.1.21 BitwiseRotateMBS(value as UInt64, count as Integer, offset as Integer, width as Integer) as UInt64	542
* 26.1.2 CompareNumbersMBS(v1 as Variant, v2 as Variant) as Integer	535
* 26.1.22 ConvertFromFloat16MBS(Number as UInt16) as Single	543
* 26.1.23 ConvertToFloat16MBS(Number as Single) as UInt16	543
* 26.1.24 CosHMBS(x as Double) as Double	544
* 26.1.25 CosMBS(x as Double) as Double	545
* 26.1.26 CurrencyAddMBS(value1 as Currency, value2 as Currency) as Currency	545
* 26.1.27 CurrencyDivMBS(value1 as Currency, value2 as Integer) as Currency	545
* 26.1.28 CurrencyMulMBS(value1 as Currency, value2 as Integer) as Currency	545
* 26.1.29 CurrencySubMBS(value1 as Currency, value2 as Currency) as Currency	546
* 26.1.30 CurrencyValueMBS(value as string) as Currency	546
* 26.1.31 DoubleToExtendedStrMBS(x as Double) as string	547
* 26.1.59 DoubleToInt64MBS(value as Double) as Int64	557
* 26.1.60 DoubleToUInt64MBS(value as Double) as UInt64	557
* 26.1.32 Exp2MBS(x as Double) as Double	547
* 26.1.33 ExpMBS(x as Double) as Double	547
* 26.1.34 ExtendedStrToDoubleMBS(v as string) as Double	548
* 26.1.35 FacMBS(x as Integer) as Double	548
* 26.1.36 FloorMBS(x as Double) as Double	549
* 26.1.37 FRExpMBS(inputx as Double, byref expValue as Integer) as Double	549

* 26.1.38 HiWordMBS(i as Integer) as Integer	550
* 26.1.39 HypotMBS(x as Double, y as Double) as Double	550
* 26.1.61 Int64ToDoubleMBS(value as Int64) as Double	558
* 26.1.40 IsFiniteMBS(x as Double) as boolean	550
* 26.1.41 IsInfMBS(x as Double) as boolean	550
* 26.1.42 IsNaNMBS(x as Double) as boolean	551
* 26.1.1 IsValidCreditCardNumberMBS(Number as String) as boolean	535
* 26.1.43 Log10MBS(x as Double) as Double	551
* 26.1.44 Log2MBS(x as Double) as Double	551
* 26.1.45 LogicalShiftMBS(value as UInt64, count as Integer) as UInt64	552
* 26.1.46 LogMBS(x as Double) as Double	552
* 26.1.47 LoWordMBS(i as Integer) as Integer	552
* 26.1.48 NormInvMBS(p as Double, mu as double = 0.0, sigma as double = 1.0) as double	553
* 26.1.49 PowMBS(x as Double, y as Double) as Double	553
* 26.1.50 RandomExponentialDistributionMBS(lambda as Double) as double	554
* 26.1.51 RandomNormalDistributionMBS(Mean as Double, StdDev as Double) as double	554
* 26.1.52 RandomPoissonDistributionMBS(Mean as Integer) as Integer	554
* 26.1.53 RoundMBS(x as Double, decimals as Integer = 0) as Double	555
* 26.1.54 SinHMBS(x as Double) as Double	555
* 26.1.55 SinMBS(x as Double) as Double	556
* 26.1.56 SqrtMBS(x as Double, y as Double) as Double	556
* 26.1.57 TanHMBS(x as Double) as Double	556
* 26.1.58 TanMBS(x as Double) as Double	557
* 26.1.62 UInt64ToDoubleMBS(value as UInt64) as Double	558

- **10 Common Types** 241
 - 10.1 Globals 241
 - * 10.1.1 MakeDoublePointMBS(x as Double, y as Double) as DoublePointMBS 241
 - * 10.1.2 MakeDoubleRectMBS(left as Double, top as Double, width as Double, height as Double) as DoubleRectMBS 241
 - * 10.1.3 MakeIntegerPointMBS(x as Integer, y as Integer) as IntegerPointMBS 241
 - * 10.1.4 MakeIntegerRectMBS(left as Integer, top as Integer, width as Integer, height as Integer) as IntegerRectMBS 242

• 26 Math	535
– ?? Globals	??
* 26.1.3 ACosHMBS(x as Double) as Double	536
* 26.1.4 ACosMBS(x as Double) as Double	536
* 26.1.5 ArithmeticShiftMBS(value as UInt64, count as Integer) as UInt64	537
* 26.1.6 ASinHMBS(x as Double) as Double	537
* 26.1.7 ASinMBS(x as Double) as Double	537
* 26.1.8 ATan2MBS(x as Double, y as Double) as Double	538
* 26.1.9 ATanHMBS(x as Double) as Double	538
* 26.1.10 ATanMBS(x as Double) as Double	539
* 26.1.11 BitClearMBS(value as UInt64, mask as UInt64) as UInt64	539
* 26.1.12 BitCountMBS(value as UInt64) as Integer	539
* 26.1.13 BitExclMBS(value as UInt64, bitNumber as Integer) as UInt64	540
* 26.1.14 BitInclMBS(value as UInt64, bitNumber as Integer) as UInt64	540
* 26.1.15 BitIsSetMBS(value as UInt64, bitNumber as Integer) as Boolean	540
* 26.1.16 BitValMBS(bitNumber as Integer) as UInt64	541
* 26.1.17 BitwiseDiffMBS(x as UInt64, y as UInt64) as UInt64	541
* 26.1.18 BitwiseNAndMBS(x as UInt64, y as UInt64) as UInt64	541
* 26.1.19 BitwiseNOrMBS(x as UInt64, y as UInt64) as UInt64	542
* 26.1.20 BitwiseNotMBS(value as UInt64) as UInt64	542
* 26.1.21 BitwiseRotateMBS(value as UInt64, count as Integer, offset as Integer, width as Integer) as UInt64	542
* 26.1.2 CompareNumbersMBS(v1 as Variant, v2 as Variant) as Integer	535
* 26.1.22 ConvertFromFloat16MBS(Number as UInt16) as Single	543
* 26.1.23 ConvertToFloat16MBS(Number as Single) as UInt16	543
* 26.1.24 CosHMBS(x as Double) as Double	544
* 26.1.25 CosMBS(x as Double) as Double	545
* 26.1.26 CurrencyAddMBS(value1 as Currency, value2 as Currency) as Currency	545
* 26.1.27 CurrencyDivMBS(value1 as Currency, value2 as Integer) as Currency	545
* 26.1.28 CurrencyMulMBS(value1 as Currency, value2 as Integer) as Currency	545
* 26.1.29 CurrencySubMBS(value1 as Currency, value2 as Currency) as Currency	546
* 26.1.30 CurrencyValueMBS(value as string) as Currency	546
* 26.1.31 DoubleToExtendedStrMBS(x as Double) as string	547
* 26.1.59 DoubleToInt64MBS(value as Double) as Int64	557
* 26.1.60 DoubleToUInt64MBS(value as Double) as UInt64	557
* 26.1.32 Exp2MBS(x as Double) as Double	547
* 26.1.33 ExpMBS(x as Double) as Double	547
* 26.1.34 ExtendedStrToDoubleMBS(v as string) as Double	548
* 26.1.35 FacMBS(x as Integer) as Double	548
* 26.1.36 FloorMBS(x as Double) as Double	549
* 26.1.37 FRExpMBS(inputx as Double, byref expValue as Integer) as Double	549

* 26.1.38 HiWordMBS(i as Integer) as Integer	550
* 26.1.39 HypotMBS(x as Double, y as Double) as Double	550
* 26.1.61 Int64ToDoubleMBS(value as Int64) as Double	558
* 26.1.40 IsFiniteMBS(x as Double) as boolean	550
* 26.1.41 IsInfMBS(x as Double) as boolean	550
* 26.1.42 IsNaNMBS(x as Double) as boolean	551
* 26.1.1 IsValidCreditCardNumberMBS(Number as String) as boolean	535
* 26.1.43 Log10MBS(x as Double) as Double	551
* 26.1.44 Log2MBS(x as Double) as Double	551
* 26.1.45 LogicalShiftMBS(value as UInt64, count as Integer) as UInt64	552
* 26.1.46 LogMBS(x as Double) as Double	552
* 26.1.47 LoWordMBS(i as Integer) as Integer	552
* 26.1.48 NormInvMBS(p as Double, mu as double = 0.0, sigma as double = 1.0) as double	553
* 26.1.49 PowMBS(x as Double, y as Double) as Double	553
* 26.1.50 RandomExponentialDistributionMBS(lambda as Double) as double	554
* 26.1.51 RandomNormalDistributionMBS(Mean as Double, StdDev as Double) as double	554
* 26.1.52 RandomPoissonDistributionMBS(Mean as Integer) as Integer	554
* 26.1.53 RoundMBS(x as Double, decimals as Integer = 0) as Double	555
* 26.1.54 SinHMBS(x as Double) as Double	555
* 26.1.55 SinMBS(x as Double) as Double	556
* 26.1.56 SqrtMBS(x as Double, y as Double) as Double	556
* 26.1.57 TanHMBS(x as Double) as Double	556
* 26.1.58 TanMBS(x as Double) as Double	557
* 26.1.62 UInt64ToDoubleMBS(value as UInt64) as Double	558

	13
• 29 Process	631
– 29.5.1 class ConsoleApplication	652
* 29.5.3 ArgumentsMBS as String()	652
* 29.5.4 LaunchTimeMBS as double	652
* 29.5.5 ProcessTimeMBS as double	653

- **34 Shell** 725
 - **?? Globals** ??
 - * 34.1.1 ConsoleExecuteMBS(path as folderitem, arguments() as string, environment() as string) as Integer 725
 - * 34.1.2 ConsoleExecuteMBS(path as string, arguments() as string, environment() as string) as Integer 725

	15
• 11 Controls	251
– ?? Globals	??
* 11.1.2 ShowModalThreadSafeMBS(extends theMessageDialog as MessageDialog)	251
* 11.1.1 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as DesktopWindow)	251
* 11.1.3 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as window)	252

• 13 CPUInfo	269
– 13.1.1 class CPUIDMBS	269
* 13.1.3 BrandString as String	270
* 13.1.4 CodeName as String	270
* 13.1.5 CPUID(Selector as Integer) as boolean	270
* 13.1.6 ExtFamily as Integer	270
* 13.1.7 ExtModel as Integer	271
* 13.1.8 Family as Integer	271
* 13.1.9 FeatureName(index as Integer) as String	271
* 13.1.10 Flags(index as Integer) as Boolean	271
* 13.1.11 L1DataCache as Integer	272
* 13.1.12 L1InstructionCache as Integer	272
* 13.1.13 L2Cache as Integer	272
* 13.1.14 L3Cache as Integer	272
* 13.1.15 Model as Integer	272
* 13.1.16 NumCores as Integer	273
* 13.1.17 NumLogicalCPUs as Integer	273
* 13.1.18 Stepping as Integer	273
* 13.1.19 TotalLogicalCPUs as Integer	273
* 13.1.20 Vendor as Integer	273
* 13.1.21 VendorName as String	274
* 13.1.23 BrandString as String	274
* 13.1.24 EAX as UInt32	274
* 13.1.25 EBX as UInt32	274
* 13.1.26 ECX as UInt32	275
* 13.1.27 EDX as UInt32	275
* 13.1.28 Family as Integer	275
* 13.1.29 Model as Integer	276
* 13.1.30 ProcessorVendor as String	276
* 13.1.31 Stepping as Integer	277

	17
• 39 System	1143
– ?? Globals	??
* 39.1.6 AbortMBS	1146
* 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()	1147
* 39.1.1 CrashNiceMBS	1143
* 39.1.2 CrashUglyMBS	1143
* 39.1.3 DelayMBS(time as Double)	1143
* 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
* 39.1.7 ExitMBS(code as Integer)	1146
* 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
* 39.1.9 MillisecondsMBS as Double	1147
* 39.1.5 SleepMBS(time as Double)	1145

• 18 Encryption and Hash	345
– ?? Globals	??
* 18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32	353
* 18.1.3 CRC16MBS(data as string) as UInt16	346
* 18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	346
* 18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32	347
* 18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32	347
* 18.1.7 CRC_32OfStrMBS(s as String) as UInt32	347
* 18.1.8 CRC_CCITInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	348
* 18.1.9 CRC_CCITInMemMBS(address as Ptr, length as Integer) as UInt32	348
* 18.1.10 CRC_CCITOfStrContMBS(s as String, prevCRC as UInt32) as UInt32	349
* 18.1.11 CRC_CCITOfStrMBS(s as String) as UInt32	349
* 18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String	349
* 18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String	349
* 18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64	349
* 18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64	352
* 18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64	350
* 18.1.21 CurrentUnixTimeMBS as UInt64	353
* 18.1.22 DecodeFromBase32MBS(data as string) as String	353
* 18.1.23 EncodeToBase32MBS(data as string) as String	354
* 18.1.16 GetHash32MBS(s as string) as UInt32	350
* 18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16	351
* 18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string	351

	19
• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock,len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock,len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222
– 8.2.1 class DateDifferenceMBS	224
* 8.2.3 Calc(StartDate as date, EndDate as date) as boolean	225
* 8.2.4 Calc(StartDate as dateTime, EndDate as dateTime) as boolean	226
* 8.2.5 Constructor	227
* 8.2.6 Constructor(StartDate as date, EndDate as date)	227
* 8.2.7 Constructor(StartDate as dateTime, EndDate as dateTime)	227
* 8.2.8 isLeapYear(year as Integer) as boolean	228
* 8.2.10 Day as Integer	228
* 8.2.11 EndDate as Variant	228
* 8.2.12 EndDay as Integer	228
* 8.2.13 EndHour as Integer	229
* 8.2.14 EndMinute as Integer	229
* 8.2.15 EndMonth as Integer	229
* 8.2.16 EndSecond as Integer	229
* 8.2.17 EndTotalSeconds as Double	229
* 8.2.18 EndYear as Integer	229
* 8.2.19 Hour as Integer	230
* 8.2.20 Minute as Integer	230
* 8.2.21 Month as Integer	230

* 8.2.22 Ready as Boolean	230
* 8.2.23 Second as Integer	230
* 8.2.24 StartDate as Variant	231
* 8.2.25 StartDay as Integer	231
* 8.2.26 StartHour as Integer	231
* 8.2.27 StartMinute as Integer	231
* 8.2.28 StartMonth as Integer	231
* 8.2.29 StartSecond as Integer	231
* 8.2.30 StartTotalSeconds as Double	232
* 8.2.31 StartYear as Integer	232
* 8.2.32 Swap as Boolean	232
* 8.2.33 TotalDay as Integer	232
* 8.2.34 TotalSeconds as Double	233
* 8.2.35 Year as Integer	233

	21
• 17 DynamicDeclares	323
– 17.1.1 class DeclareCallBackMBS	323
* 17.1.3 Constructor(Signature as String)	324
* 17.1.4 ParameterType(Index as Integer) as String	324
* 17.1.6 AllowAsync as Boolean	324
* 17.1.7 CallCount as Integer	325
* 17.1.8 FunctionPtr as Ptr	325
* 17.1.9 Name as String	325
* 17.1.10 ParameterCount as Integer	325
* 17.1.11 Signature as String	325
* 17.1.12 SignatureParameters as String	326
* 17.1.13 SignatureReturn as String	326
* 17.1.14 Tag as Variant	326
* 17.1.15 ParameterCFRetain(Index as Integer) as Boolean	326
* 17.1.16 ParameterCopyString(Index as Integer) as Boolean	326
* 17.1.17 ParameterNSRetain(Index as Integer) as Boolean	327
* 17.1.19 Callback(Parameters() as Variant) as Variant	327
– 17.2.1 class DeclareFunctionMBS	329
* 17.2.3 ClearParameters	330
* 17.2.4 Constructor(Signature as String, FunctionPtr as Ptr)	330
* 17.2.5 Invoke as Variant	330
* 17.2.6 Invoke(Parameters() as Variant) as Variant	331
* 17.2.7 SetParameters(paramArray Parameters as Variant)	331
* 17.2.8 SetParameters(Parameters() as Variant)	332
* 17.2.10 CallCount as Integer	333
* 17.2.11 CallMode as Integer	333
* 17.2.12 FunctionPtr as Ptr	333
* 17.2.13 Name as String	334
* 17.2.14 ParameterCount as Integer	334
* 17.2.15 Parameters as Dictionary	334
* 17.2.16 Signature as String	334
* 17.2.17 SignatureParameters as String	334
* 17.2.18 SignatureReturn as String	335
* 17.2.19 StackSize as Integer	335
* 17.2.20 Tag as Variant	335
* 17.2.21 ParameterBoolean(Index as Integer) as Boolean	335
* 17.2.22 ParameterDouble(Index as Integer) as Double	335
* 17.2.23 ParameterInteger(Index as Integer) as Int64	336
* 17.2.24 ParameterPointer(Index as Integer) as Ptr	336
* 17.2.25 ParameterSingle(Index as Integer) as Single	336
* 17.2.26 ParameterString(Index as Integer) as String	336

* 17.2.27 ParameterValue(Index as Integer) as Variant	337
– 17.3.1 class DeclareLibraryMBS	340
* 17.3.3 Constructor(LibFile as folderItem)	340
* 17.3.4 Constructor(LibPath as string)	340
* 17.3.5 Symbol(name as string) as ptr	341
* 17.3.6 SymbolName(index as Integer) as string	341
* 17.3.7 SymbolName(p as Ptr) as string	342
* 17.3.8 SymbolNames as string()	342
* 17.3.10 Handle as Integer	342
* 17.3.11 Name as String	342
* 17.3.12 Path as String	343
* 17.3.13 SymbolCount as Integer	343

	23
• 39 System	1143
– ?? Globals	??
* 39.1.6 AbortMBS	1146
* 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()	1147
* 39.1.1 CrashNiceMBS	1143
* 39.1.2 CrashUglyMBS	1143
* 39.1.3 DelayMBS(time as Double)	1143
* 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
* 39.1.7 ExitMBS(code as Integer)	1146
* 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
* 39.1.9 MillisecondsMBS as Double	1147
* 39.1.5 SleepMBS(time as Double)	1145

• 29 Process	631
– 29.7.1 class DesktopApplication	656
* 29.7.3 ApplicationNameMBS as string	656
* 29.7.4 ArgumentsMBS as String()	656
* 29.7.5 BundleFolderMBS as folderitem	656
* 29.7.6 BundleLocalizedStringMBS(key as string) as string	657
* 29.7.7 BundleLocalizedStringMBS(key as string,fromtable as string) as string	657
* 29.7.8 BundleResourceFolderItemLocalizedMBS(ResourceName as string, ResourceType as string, SubDirectory as string) as folderitem	658
* 29.7.9 BundleResourceFolderMBS as folderitem	658
* 29.7.10 HideMeMBS as boolean	658
* 29.7.11 HideOthersMBS as boolean	659
* 29.7.12 LaunchTimeMBS as double	659
* 29.7.13 ProcessTimeMBS as double	659
* 29.7.15 FrontmostMBS as Boolean	660

	25
• 11 Controls	251
– 11.2.1 class DesktopLabel	253
* 11.2.3 SetTextThreadSafeMBS(text as string)	253
– 11.3.1 class DesktopListbox	254
* 11.3.3 RefreshCellThreadSafeMBS(Row As Integer, Column As Integer)	254
– 11.4.1 class DesktopProgressbar	255
* 11.4.3 SetMaximumThreadSafeMBS(maximum as integer)	255
* 11.4.4 SetValueThreadSafeMBS(value as integer)	255

• 11 Controls	251
– 9.1.1 class DesktopTextArea	237
* 9.1.3 SetTextThreadSafeMBS(text as string)	238
– 11.5.1 class DesktopTextField	256
* 11.5.3 SetTextThreadSafeMBS(text as string)	256
– 11.6.1 class DesktopUIControl	257
* 11.6.3 RefreshThreadSafeMBS(immediately as Boolean = False)	257
* 11.6.4 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately as Boolean = False)	257
* 11.6.5 SetEnabledThreadSafeMBS(value as boolean)	258
* 11.6.6 SetVisibleThreadSafeMBS(value as boolean)	258

• 41 Window	1187
– 41.1.1 class DesktopWindow	1187
* 41.1.3 ActivateWindowMBS	1187
* 41.1.4 BackingScaleFactorMBS as double	1188
* 41.1.5 CleanUpTransparentMBS(refValue as integer)	1188
* 41.1.7 CollapsableMBS as Boolean	1189
* 41.1.8 ConstrainWindowToScreenMBS(animate as Boolean)	1189
* 41.1.9 IsFullScreenMBS as Boolean	1189
* 41.1.10 MakeTransparentMBS as integer	1189
* 41.1.11 RefreshThreadSafeMBS(immediately As Boolean = False)	1190
* 41.1.12 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately As Boolean = False)	1190
* 41.1.13 RemoveWindowProxyIconMBS	1191
* 41.1.14 SetTransparencyMBS(value as integer) as boolean	1191
* 41.1.15 ShowHideToolbarMBS(animate as Boolean, value as Boolean)	1192
* 41.1.16 SmoothResizeCenteredMBS(width as integer,height as integer)	1192
* 41.1.17 SmoothResizeMBS(width as integer,height as integer)	1192
* 41.1.18 ToggleFullScreenMBS as Boolean	1193
* 41.1.19 WinFlashWindowMBS(Invert as boolean)	1193
* 41.1.21 CanBeVisibleWithoutLoginMBS as Boolean	1194
* 41.1.22 CollapsedMBS as Boolean	1194
* 41.1.23 FullScreenAuxiliaryMBS as Boolean	1195
* 41.1.24 FullScreenPrimaryMBS as Boolean	1195
* 41.1.25 HasBorderMBS as Boolean	1195
* 41.1.26 HasCaptionMBS as Boolean	1195
* 41.1.27 HasCloseBoxMBS as Boolean	1196
* 41.1.28 HasCollapseBoxMBS as Boolean	1196
* 41.1.29 HasMaximizeBoxMBS as Boolean	1196
* 41.1.30 HasMinimizeBoxMBS as Boolean	1197
* 41.1.31 HasNoShadowMBS as Boolean	1197
* 41.1.32 HasNoTitleBarMBS as Boolean	1198
* 41.1.33 HasSystemMenuMBS as Boolean	1198
* 41.1.34 HasToolbarButtonMBS as Boolean	1199
* 41.1.35 IgnoreClicksMBS as Boolean	1199
* 41.1.36 IsIconicMBS as Boolean	1200
* 41.1.37 IsMetalWindowMBS as Boolean	1200
* 41.1.38 IsResizableMBS as Boolean	1201
* 41.1.39 IsZoomedMacMBS as Boolean	1201
* 41.1.40 IsZoomedMBS as Boolean	1201
* 41.1.41 ModifiedMBS as Boolean	1202
* 41.1.42 ToolbarVisibleMBS as Boolean	1202
* 41.1.43 TransparencyMBS as single	1202
* 41.1.44 UnifiedTitleAndToolbarMBS as Boolean	1203
* 41.1.45 WindowProxyIconFileMBS as folderitem	1203

- 12 **CoreGraphics** 267
 - 41.1.1 class DesktopWindow 1187
 - * 41.1.6 ClearTransparencyMBS 1189

• 41 Window	1187
– 41.1.1 class DesktopWindow	1187
* 41.1.3 ActivateWindowMBS	1187
* 41.1.4 BackingScaleFactorMBS as double	1188
* 41.1.5 CleanUpTransparentMBS(refValue as integer)	1188
* 41.1.7 CollapsableMBS as Boolean	1189
* 41.1.8 ConstrainWindowToScreenMBS(animate as Boolean)	1189
* 41.1.9 IsFullScreenMBS as Boolean	1189
* 41.1.10 MakeTransparentMBS as integer	1189
* 41.1.11 RefreshThreadSafeMBS(immediately As Boolean = False)	1190
* 41.1.12 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately As Boolean = False)	1190
* 41.1.13 RemoveWindowProxyIconMBS	1191
* 41.1.14 SetTransparencyMBS(value as integer) as boolean	1191
* 41.1.15 ShowHideToolbarMBS(animate as Boolean, value as Boolean)	1192
* 41.1.16 SmoothResizeCenteredMBS(width as integer,height as integer)	1192
* 41.1.17 SmoothResizeMBS(width as integer,height as integer)	1192
* 41.1.18 ToggleFullScreenMBS as Boolean	1193
* 41.1.19 WinFlashWindowMBS(Invert as boolean)	1193
* 41.1.21 CanBeVisibleWithoutLoginMBS as Boolean	1194
* 41.1.22 CollapsedMBS as Boolean	1194
* 41.1.23 FullScreenAuxiliaryMBS as Boolean	1195
* 41.1.24 FullScreenPrimaryMBS as Boolean	1195
* 41.1.25 HasBorderMBS as Boolean	1195
* 41.1.26 HasCaptionMBS as Boolean	1195
* 41.1.27 HasCloseBoxMBS as Boolean	1196
* 41.1.28 HasCollapseBoxMBS as Boolean	1196
* 41.1.29 HasMaximizeBoxMBS as Boolean	1196
* 41.1.30 HasMinimizeBoxMBS as Boolean	1197
* 41.1.31 HasNoShadowMBS as Boolean	1197
* 41.1.32 HasNoTitleBarMBS as Boolean	1198
* 41.1.33 HasSystemMenuMBS as Boolean	1198
* 41.1.34 HasToolbarButtonMBS as Boolean	1199
* 41.1.35 IgnoreClicksMBS as Boolean	1199
* 41.1.36 IsIconicMBS as Boolean	1200
* 41.1.37 IsMetalWindowMBS as Boolean	1200
* 41.1.38 IsResizableMBS as Boolean	1201
* 41.1.39 IsZoomedMacMBS as Boolean	1201
* 41.1.40 IsZoomedMBS as Boolean	1201
* 41.1.41 ModifiedMBS as Boolean	1202
* 41.1.42 ToolbarVisibleMBS as Boolean	1202
* 41.1.43 TransparencyMBS as single	1202
* 41.1.44 UnifiedTitleAndToolbarMBS as Boolean	1203
* 41.1.45 WindowProxyIconFileMBS as folderitem	1203

- **42 Windows** 1227
 - 41.1.1 class DesktopWindow 1187
 - * 41.1.46 WinMenuHandleMBS as integer 1203

	31
• 21 Files	379
– 21.1.1 class DirectorySizeMBS	379
* 21.1.3 Add(d as DirectorySizeMBS)	380
* 21.1.4 close	380
* 21.1.5 Constructor	380
* 21.1.6 Update(folder as folderitem, recursive as boolean, ticks as Integer) as boolean	380
* 21.1.8 Cancel as Boolean	381
* 21.1.9 CompressedSize as UInt64	381
* 21.1.10 CountBundlesAsItem as Boolean	382
* 21.1.11 Directory as FolderItem	382
* 21.1.12 FilesCount as UInt64	382
* 21.1.13 FolderCount as UInt64	383
* 21.1.14 HiddenCompressedSize as UInt64	383
* 21.1.15 HiddenFilesCount as Integer	383
* 21.1.16 HiddenFolderCount as Integer	384
* 21.1.17 HiddenItemCount as UInt64	384
* 21.1.18 HiddenLogicalDataForkSize as UInt64	384
* 21.1.19 HiddenLogicalResourceForkSize as UInt64	385
* 21.1.20 HiddenLogicalTotalSize as UInt64	385
* 21.1.21 HiddenPhysicalDataForkSize as UInt64	385
* 21.1.22 HiddenPhysicalResourceForkSize as UInt64	386
* 21.1.23 HiddenPhysicalTotalSize as UInt64	386
* 21.1.24 IgnoreHiddenFolderContent as Boolean	386
* 21.1.25 ItemCount as UInt64	386
* 21.1.26 LogicalDataForkSize as UInt64	387
* 21.1.27 LogicalResourceForkSize as UInt64	387
* 21.1.28 LogicalTotalSize as UInt64	387
* 21.1.29 PhysicalDataForkSize as UInt64	388
* 21.1.30 PhysicalResourceForkSize as UInt64	388
* 21.1.31 PhysicalTotalSize as UInt64	388
* 21.1.32 QueryCompressedSizes as Boolean	389
* 21.1.33 RecursionLimit as Integer	389
* 21.1.34 RecursionMaxLevel as Integer	389
* 21.1.35 VisibleCompressedSize as UInt64	389
* 21.1.36 VisibleFilesCount as Integer	390
* 21.1.37 VisibleFolderCount as Integer	390
* 21.1.38 VisibleItemCount as UInt64	390
* 21.1.39 VisibleLogicalDataForkSize as UInt64	391
* 21.1.40 VisibleLogicalResourceForkSize as UInt64	391
* 21.1.41 VisibleLogicalTotalSize as UInt64	391
* 21.1.42 VisiblePhysicalDataForkSize as UInt64	392
* 21.1.43 VisiblePhysicalResourceForkSize as UInt64	392
* 21.1.44 VisiblePhysicalTotalSize as UInt64	392
* 21.1.45 YieldTicks as Integer	393

- 16 Disassembler 321
 - 16.1 Globals 321
 - * 16.1.1 DisAssembleMBS 321
 - * 16.1.2 DisAssembleObjectMethodMBS(target as object, Declaration as string) 321
 - * 16.1.3 GetDisAssembleMBS as string 322

	33
• 32 Resolution	703
– 32.1.1 class DisplayMBS	703
* 32.1.3 CanDepth(depth as Integer) As Boolean	704
* 32.1.4 FadeGamma(intensity as Integer, col As Color)	704
* 32.1.5 FadeGammaTo(intensity as Integer, col As Color, ticks as Integer)	704
* 32.1.6 GetBestResolution(width as Integer, height as Integer, depth as Integer, safe As Boolean) As ResolutionMBS	704
* 32.1.7 GetBestResolution(width as Integer, height as Integer, safe As Boolean) As ResolutionMBS	705
* 32.1.8 GetCurrentResolution As ResolutionMBS	705
* 32.1.9 GetLargestResolution(depth as Integer, safe As Boolean) As ResolutionMBS	705
* 32.1.10 GetLargestResolution(safe As Boolean) As ResolutionMBS	705
* 32.1.11 GetResolution(num as Integer) As ResolutionMBS	706
* 32.1.12 ResolutionCount(depth as Integer, safe As Boolean) as Integer	706
* 32.1.13 ResolutionCount(safe As Boolean) as Integer	706
* 32.1.14 SetDepth(depth as Integer) As Boolean	706
* 32.1.15 SwitchTo(width as Integer, height as Integer, depth as Integer, safe As Boolean) As Boolean	707
* 32.1.16 Update	707
* 32.1.18 Depth as Integer	707
* 32.1.19 displaynum as Integer	707
* 32.1.20 GammaColor as color	708
* 32.1.21 GammaIntensity as Integer	708
* 32.1.22 Height as Integer	708
* 32.1.23 hz as Integer	708
* 32.1.24 Left as Integer	709
* 32.1.25 Top as Integer	709
* 32.1.26 Width as Integer	709
* 32.1.27 NativeGamma as memoryblock	709
– ?? Globals	??
* 32.2.1 DisplayCountMBS as Integer	711
* 32.2.2 GetDisplayMBS(num as Integer) As DisplayMBS	711
* 32.2.3 ResolutionLibraryPresentMBS as boolean	711
* 32.2.4 UpdateDisplayCountMBS	712

• 10 Common Types	241
– 10.2.1 class DoublePointMBS	242
* 10.2.3 Move(deltax as Double, deltay as Double)	242
* 10.2.5 x as Double	242
* 10.2.6 y as Double	242
– 10.3.1 class DoubleRectMBS	243
* 10.3.3 Intersection(other as DoubleRectMBS) as DoubleRectMBS	243
* 10.3.4 Intersects(other as DoubleRectMBS) as boolean	243
* 10.3.5 Move(deltax as Double, deltay as Double)	243
* 10.3.7 Bottom as Double	243
* 10.3.8 height as Double	244
* 10.3.9 left as Double	244
* 10.3.10 right as Double	244
* 10.3.11 Size as Double	244
* 10.3.12 top as Double	244
* 10.3.13 width as Double	244

• 19 Endian	359
– 19.1 Globals	359
* 19.1.1 EndianS16_BtoLMBS(n as Int16) as Int16	359
* 19.1.2 EndianS16_BtoNMBS(n as Int16) as Int16	359
* 19.1.3 EndianS16_LtoBMBS(n as Int16) as Int16	360
* 19.1.4 EndianS16_LtoNMBS(n as Int16) as Int16	360
* 19.1.5 EndianS16_NtoBMBS(n as Int16) as Int16	360
* 19.1.6 EndianS16_NtoLMBS(n as Int16) as Int16	361
* 19.1.7 EndianS32_BtoLMBS(n as Int32) as Int32	361
* 19.1.8 EndianS32_BtoNMBS(n as Int32) as Int32	361
* 19.1.9 EndianS32_LtoBMBS(n as Int32) as Int32	362
* 19.1.10 EndianS32_LtoNMBS(n as Int32) as Int32	362
* 19.1.11 EndianS32_NtoBMBS(n as Int32) as Int32	362
* 19.1.12 EndianS32_NtoLMBS(n as Int32) as Int32	363
* 19.1.13 EndianSwap16MBS(n as UInt16) as UInt16	363
* 19.1.14 EndianSwap32MBS(n as UInt32) as UInt32	363
* 19.1.15 EndianU16_BtoLMBS(n as UInt16) as UInt16	363
* 19.1.16 EndianU16_BtoNMBS(n as UInt16) as UInt16	364
* 19.1.17 EndianU16_LtoBMBS(n as UInt16) as UInt16	364
* 19.1.18 EndianU16_LtoNMBS(n as UInt16) as UInt16	364
* 19.1.19 EndianU16_NtoBMBS(n as UInt16) as UInt16	365
* 19.1.20 EndianU16_NtoLMBS(n as UInt16) as UInt16	365
* 19.1.21 EndianU32_BtoLMBS(n as UInt32) as UInt32	365
* 19.1.22 EndianU32_BtoNMBS(n as UInt32) as UInt32	366
* 19.1.23 EndianU32_LtoBMBS(n as UInt32) as UInt32	366
* 19.1.24 EndianU32_LtoNMBS(n as UInt32) as UInt32	366
* 19.1.25 EndianU32_NtoBMBS(n as UInt32) as UInt32	367
* 19.1.26 EndianU32_NtoLMBS(n as UInt32) as UInt32	367

• 29 Process	631
– 29.8.1 class EnvironmentMBS	661
* 29.8.3 Add(name as string,value as string) as boolean	661
* 29.8.4 Get(name as string) as string	661
* 29.8.5 Lines as string()	662
* 29.8.6 Name(Index as Integer) as string	662
* 29.8.7 Names as string()	662
* 29.8.8 Update	662
* 29.8.10 Count as Integer	662
* 29.8.11 Value(Index as Integer) as string	663

	37
• 21 Files	379
– ?? Globals	??
* 21.2.2 AdminToolsMBS(domain as Integer) as folderitem	394
* 21.2.3 CookiesMBS as folderitem	395
* 21.2.1 ExchangeFilesMBS(first as folderitem, second as folderitem) as Integer	394
* 21.2.4 HistoryMBS as folderitem	395
* 21.2.5 InternetCacheMBS as folderitem	395
* 21.2.7 SetCurrentWorkingDirectoryMBS(path as folderitem) as boolean	395
* 21.2.8 SetDllDirectoryMBS(path as folderitem) as boolean	396
* 21.2.9 VolumeFreeSizePathMBS(Path as String) as Int64	397
* 21.2.10 VolumeSizePathMBS(Path as String) as Int64	397
* 21.2.6 WindowsStartMenuMBS(domain as Integer) as folderitem	395

• 39 System	1143
– ?? Globals	??
* 39.1.6 AbortMBS	1146
* 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()	1147
* 39.1.1 CrashNiceMBS	1143
* 39.1.2 CrashUglyMBS	1143
* 39.1.3 DelayMBS(time as Double)	1143
* 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
* 39.1.7 ExitMBS(code as Integer)	1146
* 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
* 39.1.9 MillisecondsMBS as Double	1147
* 39.1.5 SleepMBS(time as Double)	1145

	39
• 21 Files	379
– 21.3.1 class FileListMBS	397
* 21.3.3 AttributeModificationDate(index as Integer) as Double	398
* 21.3.4 AttributeModificationDate(index as Integer, UTC as boolean) as Date	398
* 21.3.5 AttributeModificationDateTime(index as integer, UTC as boolean) as DateTime	398
* 21.3.6 BackupDate(index as Integer) as Double	399
* 21.3.7 BackupDate(index as Integer, UTC as boolean) as Date	399
* 21.3.8 BackupDateTime(index as integer, UTC as boolean) as DateTime	399
* 21.3.9 CFURL(index as integer) as Variant	399
* 21.3.10 Close	399
* 21.3.11 Constructor	400
* 21.3.12 Constructor(filelist as FileListMBS, index as Integer, WinFilter as string = "", SkipMode as Integer = 0)	400
* 21.3.13 Constructor(folder as folderitem, WinFilter as string = "", SkipMode as Integer = 0)	401
* 21.3.14 Constructor(Path as String, WinFilter as string = "", SkipMode as Integer = 0)	401
* 21.3.15 CreationDate(index as Integer) as Double	402
* 21.3.16 CreationDate(index as Integer, UTC as boolean) as Date	402
* 21.3.17 CreationDateTime(index as integer, UTC as boolean) as DateTime	402
* 21.3.18 Creator(index as Integer) as string	402
* 21.3.19 Directory(index as Integer) as boolean	403
* 21.3.20 DisplayName(index as Integer) as string	403
* 21.3.21 FinderFlags(index as Integer) as Integer	403
* 21.3.22 FSRef(index as Integer) as memoryblock	403
* 21.3.23 HFSUniStr255(index as Integer) as memoryblock	404
* 21.3.24 IsBundle(index as Integer) as Boolean	404
* 21.3.25 IsHardLinked(index as Integer) as boolean	405
* 21.3.26 Item(index as Integer) as folderitem	405
* 21.3.27 ItemPath(index as Integer) as string	405
* 21.3.28 LastAccessDate(index as Integer) as Double	405
* 21.3.29 LastAccessDate(index as Integer, UTC as boolean) as Date	406
* 21.3.30 LastAccessDateTime(index as integer, UTC as boolean) as DateTime	406
* 21.3.31 LogicalDataLength(index as Integer) as Int64	406
* 21.3.32 LogicalResourceLength(index as Integer) as Int64	406
* 21.3.33 ModificationDate(index as Integer) as Double	407
* 21.3.34 ModificationDate(index as Integer, UTC as boolean) as Date	407
* 21.3.35 ModificationDateTime(index as integer, UTC as boolean) as DateTime	407
* 21.3.36 Name(index as Integer) as string	407
* 21.3.37 NodeID(index as Integer) as Int64	407
* 21.3.38 ParentDirectoryID(index as Integer) as Int64	408
* 21.3.39 PhysicalDataLength(index as Integer) as Int64	408

* 21.3.40 PhysicalResourceLength(index as Integer) as Int64	408
* 21.3.41 SortByCreationDate	408
* 21.3.42 SortByFileName	408
* 21.3.43 SortByModificationDate	409
* 21.3.44 TrueItem(index as Integer) as folderitem	409
* 21.3.45 Type(index as Integer) as string	409
* 21.3.46 Visible(index as Integer) as boolean	409
* 21.3.47 WinFileAttributes(index as Integer) as Integer	410
* 21.3.49 Cancel as Boolean	410
* 21.3.50 Count as Integer	410
* 21.3.51 Folder as FolderItem	410
* 21.3.52 OK as Boolean	410
* 21.3.53 Path as String	411
* 21.3.54 Threaded as Boolean	411
* 21.3.55 TotalLogicalDataLength as Int64	411
* 21.3.56 TotalLogicalResourceLength as Int64	411
* 21.3.57 TotalPhysicalDataLength as Int64	411
* 21.3.58 TotalPhysicalResourceLength as Int64	412
* 21.3.59 YieldTicks as Integer	412

	41
• 20 Filemapping and Shared Memory	369
– 20.1.1 class FileMappingMBS	369
* 20.1.3 CloseFile	370
* 20.1.4 CloseFileMapping	370
* 20.1.5 Constructor	370
* 20.1.6 Constructor(file as folderitem, write as boolean = false)	371
* 20.1.7 CreateSharedMemory(name as string, Size as Int64) as boolean	371
* 20.1.8 DeleteSharedMemory(name as string) as boolean	371
* 20.1.9 EnlargeFile(Size as Int64)	372
* 20.1.10 GetSharedMemoryValue(name as string) as MemoryBlock	372
* 20.1.11 HasSharedMemoryValue(name as string) as Boolean	372
* 20.1.12 MapView(mem as MemoryBlock, offset as Int64, Size as Integer) as FileMapViewMBS	372
* 20.1.13 MapView(offset as Int64, Size as Integer) as FileMapViewMBS	373
* 20.1.14 OpenFileMapping(MaxSize as Int64 = 0) as boolean	373
* 20.1.15 OpenSharedMemory(name as string) as boolean	373
* 20.1.16 SetSharedMemoryValue(name as string, data as MemoryBlock) as Boolean	374
* 20.1.17 ShrinkFile	374
* 20.1.19 DeleteFileOnClose as Boolean	374
* 20.1.20 DeleteSharedMemory as Boolean	374
* 20.1.21 File as FolderItem	375
* 20.1.22 isWriteable as Boolean	375
* 20.1.23 Lasterror as Integer	375
* 20.1.24 LasterrorString as String	375
* 20.1.25 Name as String	376
* 20.1.26 SharedMemorySize as Int64	376
* 20.1.27 ShrinkFileOnClose as Boolean	376
– 20.2.1 class FileMapViewMBS	377
* 20.2.3 FlushView	377
* 20.2.4 UnmapView	377
* 20.2.6 FlushOnClose as Boolean	377
* 20.2.7 Memory as Memoryblock	378
* 20.2.8 Offset as Int64	378
* 20.2.9 Parent as FileMappingMBS	378
* 20.2.10 Size as Integer	378

• 21 Files	379
– 21.4.1 class FolderItem	413
* 21.4.3 CalculateDirectorySizeMBS(recursive as boolean = false, ticks as Integer = 0, QueryCompressed-Sizes as boolean = false, RecursionLimit as Integer = -1) as DirectorySizeMBS	413
* 21.4.4 CompressedFileLengthMBS as int64	413
* 21.4.5 CreateLargeBinaryStreamMBS(MacType as string, MacCreator as string) as LargeBinaryStreamMBS	414
* 21.4.6 CreateResStreamMBS(MacType as string, MacCreator as string) as ResStreamMBS	414
* 21.4.7 DeleteDataForkMBS	414
* 21.4.8 DeleteResourceForkMBS	415
* 21.4.9 DisplayPathMBS(delimiter as string = "/") as string	415
* 21.4.10 FilesMBS as FolderItem()	415
* 21.4.11 FoldersMBS as FolderItem()	416
* 21.4.14 IsCompressedFileMBS as Boolean	418
* 21.4.15 IsEncryptedFileMBS as Boolean	419
* 21.4.16 IsFileDataForkOpenReadWriteMBS as boolean	419
* 21.4.17 IsFileResourceForkOpenReadWriteMBS as boolean	419
* 21.4.18 ItemsMBS as FolderItem()	419
* 21.4.19 LogicalFileDataLengthMBS as int64	420
* 21.4.20 LogicalFileResLengthMBS as int64	420
* 21.4.21 LogicalFileTotalLengthMBS as int64	421
* 21.4.22 NameExtensionMBS as string	421
* 21.4.23 NameWithoutExtensionMBS as string	422
* 21.4.24 OpenAsLargeBinaryStreamMBS(write as Boolean) as LargeBinaryStreamMBS	422
* 21.4.25 OpenAsResStreamMBS(write as Boolean) as ResStreamMBS	423
* 21.4.26 ParentVolumeMBS as folderitem	423
* 21.4.27 PhysicalFileDataLengthMBS as int64	423
* 21.4.28 PhysicalFileResLengthMBS as int64	423
* 21.4.29 PhysicalFileTotalLengthMBS as int64	424
* 21.4.30 ReadFileMBS(byref data as MemoryBlock, offset As Integer = 0, byteCount As Integer = -1) as boolean	424
* 21.4.31 ReadFileMBS(byref data as string, offset As Integer = 0, byteCount As Integer = -1) as boolean	425
* 21.4.32 SortedFilesMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()	426
* 21.4.33 SortedFoldersMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()	426
* 21.4.34 SortedItemsMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()	427
* 21.4.35 TrueFilesMBS as FolderItem()	427
* 21.4.36 TrueFoldersMBS as FolderItem()	428

	43
* 21.4.37 TrueItemsMBS as FolderItem()	428
* 21.4.38 VolumeFreeSizeKBMBS as Int64	428
* 21.4.39 VolumeFreeSizeMBS as Int64	429
* 21.4.40 VolumeSizeKBMBS as Int64	430
* 21.4.41 VolumeSizeMBS as Int64	430
* 21.4.42 VolumeUUIDMBS as string	431
* 21.4.43 WriteFileMBS(data as MemoryBlock) as boolean	431
* 21.4.44 WriteFileMBS(data as string) as boolean	432
* 21.4.46 AccessDateMBS(UTC as boolean = false) as date	433
* 21.4.47 AccessDateTimeMBS(UTC as boolean = false) as DateTime	433
* 21.4.48 AddedToDirectoryDateMBS as date	433
* 21.4.49 AddedToDirectoryDateTimeMBS as DateTime	433
* 21.4.50 AttributeModificationDateMBS(UTC as boolean = false) as date	434
* 21.4.51 AttributeModificationDateTimeMBS(UTC as boolean = false) as DateTime	434
* 21.4.52 BackupDateMBS(UTC as boolean = false) as date	434
* 21.4.53 BackupDateTimeMBS(UTC as boolean = false) as DateTime	435
* 21.4.54 CreationDateMBS(UTC as boolean = false) as date	435
* 21.4.55 CreationDateTimeMBS(UTC as boolean = false) as DateTime	435
* 21.4.56 ModificationDateMBS(UTC as boolean = false) as date	435
* 21.4.57 ModificationDateTimeMBS(UTC as boolean = false) as DateTime	436

- **22 Fonts** 457
 - 21.4.1 class FolderItem 413
 - * 21.4.12 FontActivateMBS(OnlyLocal as boolean) as Integer 416
 - * 21.4.13 FontDeactivateMBS(OnlyLocal as boolean) as Integer 418

	45
• 21 Files	379
– 21.4.1 class FolderItem	413
* 21.4.3 CalculateDirectorySizeMBS(recursive as boolean = false, ticks as Integer = 0, QueryCompressed-Sizes as boolean = false, RecursionLimit as Integer = -1) as DirectorySizeMBS	413
* 21.4.4 CompressedFileLengthMBS as int64	413
* 21.4.5 CreateLargeBinaryStreamMBS(MacType as string, MacCreator as string) as LargeBinaryStreamMBS	414
* 21.4.6 CreateResStreamMBS(MacType as string, MacCreator as string) as ResStreamMBS	414
* 21.4.7 DeleteDataForkMBS	414
* 21.4.8 DeleteResourceForkMBS	415
* 21.4.9 DisplayPathMBS(delimiter as string = "/") as string	415
* 21.4.10 FilesMBS as FolderItem()	415
* 21.4.11 FoldersMBS as FolderItem()	416
* 21.4.14 IsCompressedFileMBS as Boolean	418
* 21.4.15 IsEncryptedFileMBS as Boolean	419
* 21.4.16 IsFileDataForkOpenReadWriteMBS as boolean	419
* 21.4.17 IsFileResourceForkOpenReadWriteMBS as boolean	419
* 21.4.18 ItemsMBS as FolderItem()	419
* 21.4.19 LogicalFileDataLengthMBS as int64	420
* 21.4.20 LogicalFileResLengthMBS as int64	420
* 21.4.21 LogicalFileTotalLengthMBS as int64	421
* 21.4.22 NameExtensionMBS as string	421
* 21.4.23 NameWithoutExtensionMBS as string	422
* 21.4.24 OpenAsLargeBinaryStreamMBS(write as Boolean) as LargeBinaryStreamMBS	422
* 21.4.25 OpenAsResStreamMBS(write as Boolean) as ResStreamMBS	423
* 21.4.26 ParentVolumeMBS as folderitem	423
* 21.4.27 PhysicalFileDataLengthMBS as int64	423
* 21.4.28 PhysicalFileResLengthMBS as int64	423
* 21.4.29 PhysicalFileTotalLengthMBS as int64	424
* 21.4.30 ReadFileMBS(byref data as MemoryBlock, offset As Integer = 0, byteCount As Integer = -1) as boolean	424
* 21.4.31 ReadFileMBS(byref data as string, offset As Integer = 0, byteCount As Integer = -1) as boolean	425
* 21.4.32 SortedFilesMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()	426
* 21.4.33 SortedFoldersMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()	426
* 21.4.34 SortedItemsMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()	427
* 21.4.35 TrueFilesMBS as FolderItem()	427
* 21.4.36 TrueFoldersMBS as FolderItem()	428

* 21.4.37 TrueItemsMBS as FolderItem()	428
* 21.4.38 VolumeFreeSizeKBMBS as Int64	428
* 21.4.39 VolumeFreeSizeMBS as Int64	429
* 21.4.40 VolumeSizeKBMBS as Int64	430
* 21.4.41 VolumeSizeMBS as Int64	430
* 21.4.42 VolumeUUIDMBS as string	431
* 21.4.43 WriteFileMBS(data as MemoryBlock) as boolean	431
* 21.4.44 WriteFileMBS(data as string) as boolean	432
* 21.4.46 AccessDateMBS(UTC as boolean = false) as date	433
* 21.4.47 AccessDateTimeMBS(UTC as boolean = false) as DateTime	433
* 21.4.48 AddedToDirectoryDateMBS as date	433
* 21.4.49 AddedToDirectoryDateTimeMBS as DateTime	433
* 21.4.50 AttributeModificationDateMBS(UTC as boolean = false) as date	434
* 21.4.51 AttributeModificationDateTimeMBS(UTC as boolean = false) as DateTime	434
* 21.4.52 BackupDateMBS(UTC as boolean = false) as date	434
* 21.4.53 BackupDateTimeMBS(UTC as boolean = false) as DateTime	435
* 21.4.54 CreationDateMBS(UTC as boolean = false) as date	435
* 21.4.55 CreationDateTimeMBS(UTC as boolean = false) as DateTime	435
* 21.4.56 ModificationDateMBS(UTC as boolean = false) as date	435
* 21.4.57 ModificationDateTimeMBS(UTC as boolean = false) as DateTime	436
– ?? Globals	??
* 21.2.2 AdminToolsMBS(domain as Integer) as folderitem	394
* 21.2.3 CookiesMBS as folderitem	395
* 21.2.1 ExchangeFilesMBS(first as folderitem, second as folderitem) as Integer	394
* 21.2.4 HistoryMBS as folderitem	395
* 21.2.5 InternetCacheMBS as folderitem	395
* 21.2.7 SetCurrentWorkingDirectoryMBS(path as folderitem) as boolean	395
* 21.2.8 SetDllDirectoryMBS(path as folderitem) as boolean	396
* 21.2.9 VolumeFreeSizePathMBS(Path as String) as Int64	397
* 21.2.10 VolumeSizePathMBS(Path as String) as Int64	397
* 21.2.6 WindowsStartMenuMBS(domain as Integer) as folderitem	395

	47
• 23 Graphics & Pictures	459
– 23.1.1 class Graphics	459
* 23.1.3 DrawRotatedTextMBS(Rotation as Double, text as string, x as Integer, y as Integer, Center as Boolean = false, alpha as Double = 1.0, NoSwapY as boolean = false, FontWidth as Integer = 0)	459
* 23.1.4 MeasureRotatedTextMBS(text as string, byref Width as Double, byref Height as Double, FontWidth as Integer = 0) as Boolean	461

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock,len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock,len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	49
• 24 Hotkey	463
– 24.1.1 class HotKeyMBS	463
* 24.1.3 Close	464
* 24.1.4 Constructor(KeyCode as Integer, Modifiers as Integer, Exclusive as Boolean = false)	464
* 24.1.5 KeyCodeForText(name as string) as Integer	464
* 24.1.7 Exclusive as Boolean	464
* 24.1.8 Handle as Integer	465
* 24.1.9 ID as Integer	465
* 24.1.10 KeyCode as Integer	465
* 24.1.11 Modifiers as Integer	465
* 24.1.13 KeyDown	465
* 24.1.14 KeyUp	466

• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

* 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
* 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
* 38.1.60 StringORMBS(a as string,b as string) as string	1124
* 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.63 StrMBS(d as Double) as string	1126
* 38.1.64 UnicodeStringMBS(s as string) as string	1126

	53
• 10 Common Types	241
– 10.4.1 class IntegerPointMBS	246
* 10.4.3 Move(deltax as Integer, deltay as Integer)	246
* 10.4.5 x as Integer	246
* 10.4.6 y as Integer	246
– 10.5.1 class IntegerRectMBS	247
* 10.5.3 Intersection(other as IntegerRectMBS) as IntegerRectMBS	247
* 10.5.4 Intersects(other as IntegerRectMBS) as boolean	247
* 10.5.5 Move(deltax as Integer, deltay as Integer)	247
* 10.5.7 bottom as Integer	247
* 10.5.8 height as Integer	248
* 10.5.9 left as Integer	248
* 10.5.10 right as Integer	248
* 10.5.11 Size as Integer	248
* 10.5.12 top as Integer	248
* 10.5.13 width as Integer	248

• 25 JavaScript Object Notation	467
– 25.1.1 class JSONEntryMBS	467
* 25.1.3 Constructor	468
* 25.1.4 Constructor(other as JSONEntryMBS)	468
* 25.1.6 Item as Variant	468
* 25.1.7 Name as String	468
– 25.2.1 class JSONIteratorMBS	469
* 25.2.3 Constructor	469
* 25.2.4 Iterator as Iterator	469
* 25.2.5 MoveNext as Boolean	470
* 25.2.6 Value as Variant	470
* 25.2.8 Index as Integer	470
* 25.2.9 Root as JSONMBS	470
– 25.3.1 class JSONMBS	472
* 25.3.3 Add(Value as Variant)	474
* 25.3.4 AddAt(index As Integer, value As Variant)	474
* 25.3.5 AddItemToArray(item as JSONMBS)	475
* 25.3.6 AddItemToObject(label as string, value as JSONMBS)	476
* 25.3.7 AddOrReplaceItemToObject(label as string, value as JSONMBS)	476
* 25.3.8 Append(Value as Variant)	477
* 25.3.9 ApplyMergePatch(target as JSONMBS, patch as JSONMBS) as JSONMBS	477
* 25.3.10 ApplyPatch(target as JSONMBS, patch as JSONMBS) as JSONMBS	478
* 25.3.11 ArrayItem(index as integer, Clone as Boolean = false) as JSONMBS	479
* 25.3.12 ArrayItems(Clone as Boolean = false) as JSONMBS()	479
* 25.3.13 Clear	479
* 25.3.14 Clone as JSONMBS	479
* 25.3.15 Close	480
* 25.3.16 Compare(Other as JSONMBS) as Integer	480
* 25.3.17 Constructor	480
* 25.3.18 Constructor(dic As Dictionary)	481
* 25.3.19 Constructor(JSONString as String)	481
* 25.3.20 Constructor(other as JSONMBS)	482
* 25.3.21 Convert as Variant	482
* 25.3.22 Convert(value as variant) as JSONMBS	483
* 25.3.23 DeleteItem(index as Integer)	484
* 25.3.24 DeleteItem(label as string)	484
* 25.3.25 Entries as JSONEntryMBS()	485
* 25.3.26 Equals(Other as JSONMBS) as Boolean	486
* 25.3.27 FilterObjectArray(Name as String, Other as JSONMBS) as JSONMBS	486
* 25.3.28 FindValueInArray(Other as JSONMBS, StartIndex as Integer = 0) as Integer	487

* 25.3.29 FindValueInObjectArray(Name as String, Other as JSONMBS, StartIndex as Integer = 0) as Integer	488
* 25.3.30 Flatten(value as JSONMBS) as JSONMBS	488
* 25.3.31 HasChild(label as string) as Boolean	489
* 25.3.32 HasKey(Key as string) as boolean	489
* 25.3.33 HasName(Name as string) as boolean	490
* 25.3.34 Insert(index as integer, value as variant)	490
* 25.3.35 isValidJSON(JSON as string) as Boolean	491
* 25.3.36 Iterate as JSONIteratorMBS	491
* 25.3.37 IterateEntries as JSONIteratorMBS	492
* 25.3.38 IterateValues as JSONIteratorMBS	492
* 25.3.39 JSONObjectCount as Integer	493
* 25.3.40 KeyAt(index As Integer) as String	493
* 25.3.41 Keys as String()	493
* 25.3.42 Load(JSONString as String)	494
* 25.3.43 Lookup(Key As String, defaultValue As Variant = nil) as Variant	494
* 25.3.44 Merge(Other as JSONMBS)	494
* 25.3.45 MergePatchFromDiff(source as JSONMBS, target as JSONMBS) as JSONMBS	495
* 25.3.46 Name(index As Integer) as String	496
* 25.3.47 NameAt(index As Integer) as String	496
* 25.3.48 Names as String()	497
* 25.3.49 NewArrayNode as JSONMBS	497
* 25.3.50 NewBooleanNode(value as Boolean) as JSONMBS	498
* 25.3.51 NewBoolNode(value as boolean) as JSONMBS	498
* 25.3.52 NewByteStringNode(Bytes as MemoryBlock) as JSONMBS	498
* 25.3.53 NewByteStringNode(Bytes as ptr, Length as UInt64) as JSONMBS	498
* 25.3.54 NewByteStringNode(Bytes as String) as JSONMBS	499
* 25.3.55 NewCurrencyNode(value as Currency) as JSONMBS	499
* 25.3.56 NewDoubleArray(values() as Double) as JSONMBS	500
* 25.3.57 NewFalseNode as JSONMBS	500
* 25.3.58 NewInt32Array(values() as Int32) as JSONMBS	500
* 25.3.59 NewInt64Array(values() as Int64) as JSONMBS	501
* 25.3.60 NewInt64Node(value as Int64) as JSONMBS	501
* 25.3.61 NewIntegerArray(values() as Integer) as JSONMBS	501
* 25.3.62 NewNullNode as JSONMBS	501
* 25.3.63 NewNumberNode(value as Double) as JSONMBS	502
* 25.3.64 NewNumberNode(value as string) as JSONMBS	502
* 25.3.65 NewObjectNode as JSONMBS	502
* 25.3.66 NewStringArray(values() as string) as JSONMBS	503
* 25.3.67 NewStringNode(value as string) as JSONMBS	503
* 25.3.68 NewTrueNode as JSONMBS	504
* 25.3.69 NewUInt32Array(values() as UInt32) as JSONMBS	504

* 25.3.70	NewUInt64Array(values() as UInt64) as JSONMBS	504
* 25.3.71	NewUInt64Node(value as UInt64) as JSONMBS	504
* 25.3.72	Operator_Compare(Other as JSONMBS) as Integer	505
* 25.3.73	Operator_Convert as Variant()	505
* 25.3.74	Operator_Convert(dic As Dictionary)	506
* 25.3.75	PatchFromDiff(source as JSONMBS, target as JSONMBS) as JSONMBS	506
* 25.3.76	PatchFromDiff(source as JSONMBS, target as JSONMBS, KeyToCopy as String) as JSONMBS	507
* 25.3.77	Query(Path as string, Options as Integer = 0) as JSONMBS	507
* 25.3.78	Remove(Index as Integer)	509
* 25.3.79	Remove(Key as string)	509
* 25.3.80	RemoveAt(Index as Integer)	509
* 25.3.81	Replace(Path as string, NewValue as Variant) as JSONMBS	510
* 25.3.82	Search(Path as string) as JSONMBS	511
* 25.3.83	Sort(Reverse as boolean = false)	512
* 25.3.84	ToHTML(NoHeader as boolean = false, CSS as string = "") as String	512
* 25.3.85	toString(formatted as boolean) as string	513
* 25.3.86	Unflatten(value as JSONMBS) as JSONMBS	513
* 25.3.87	Values as Variant()	514
* 25.3.89	ArraySize as Integer	514
* 25.3.90	ByteStringEncoding as Integer	514
* 25.3.91	CaseSensitive as Boolean	515
* 25.3.92	ChildNode as JSONMBS	515
* 25.3.93	Compact as Boolean	516
* 25.3.94	Count as Integer	516
* 25.3.95	Handle as Integer	517
* 25.3.96	IsArray as Boolean	517
* 25.3.97	IsBoolean as Boolean	517
* 25.3.98	IsEmpty as Boolean	517
* 25.3.99	IsInt32 as Boolean	518
* 25.3.100	IsInt64 as Boolean	519
* 25.3.101	IsNull as Boolean	519
* 25.3.102	IsNumber as Boolean	519
* 25.3.103	IsObject as Boolean	520
* 25.3.104	IsUInt32 as Boolean	521
* 25.3.105	IsUInt64 as Boolean	521
* 25.3.106	LastChildNode as JSONMBS	522
* 25.3.107	LastRowIndex as Integer	522
* 25.3.108	LineLengthLimit as Integer	523
* 25.3.109	Name as String	523
* 25.3.110	NewLineCharacters as String	524
* 25.3.111	NextNode as JSONMBS	524

	57
* 25.3.112 PreviousNode as JSONMBS	525
* 25.3.113 Root as JSONMBS	525
* 25.3.114 Tag as Integer	525
* 25.3.115 toString as String	526
* 25.3.116 Type as Integer	526
* 25.3.117 TypeName as String	526
* 25.3.118 Valid as Boolean	527
* 25.3.119 Value as Variant	527
* 25.3.120 ValueBoolean as Boolean	527
* 25.3.121 ValueByteString as MemoryBlock	528
* 25.3.122 ValueDouble as Double	528
* 25.3.123 ValueInt64 as Int64	528
* 25.3.124 ValueInteger as Integer	529
* 25.3.125 ValueString as String	529
* 25.3.126 ValueUInt64 as UInt64	529
* 25.3.127 Child(index As Integer) as JSONMBS	529
* 25.3.128 Child(Key As String) as JSONMBS	530
* 25.3.129 ChildAt(index As Integer) as JSONMBS	530
* 25.3.130 Operator_Subscript(index As Integer) as Variant	531
* 25.3.131 Value(index As Integer) as Variant	531
* 25.3.132 Value(Key As String) as Variant	532
* 25.3.133 ValueAt(index As Integer) as Variant	532

- **11 Controls** 251
 - 11.7.1 class Label 259
 - * 11.7.3 SetTextThreadSafeMBS(text as string) 259

	59
• 21 Files	379
– 21.5.1 class LargeBinaryStreamMBS	437
* 21.5.3 Allocate(count as int64, flags as Integer) as int64	437
* 21.5.4 close	438
* 21.5.5 Create(file as folderitem, MacType as string, MacCreator as string) as LargeBinaryStreamMBS	438
* 21.5.6 Create(path as string, MacType as string, MacCreator as string, WinShareMode as Integer = 0) as LargeBinaryStreamMBS	438
* 21.5.7 CreateResStream(file as folderitem, MacType as string, MacCreator as string) as ResStreamMBS	439
* 21.5.8 CreateResStream(path as string, MacType as string, MacCreator as string) as ResStreamMBS	439
* 21.5.9 DeleteDataFork(file as folderitem)	439
* 21.5.10 DeleteResourceFork(file as folderitem)	440
* 21.5.11 Flush	440
* 21.5.12 LockFileExclusive as boolean	440
* 21.5.13 Open(file as folderitem, write as Boolean) as LargeBinaryStreamMBS	440
* 21.5.14 Open(path as string, write as Boolean, WinShareMode as Integer = 0) as LargeBinaryStreamMBS	441
* 21.5.15 OpenAsResStream(file as folderitem, write as Boolean) as ResStreamMBS	441
* 21.5.16 OpenAsResStream(path as string, write as Boolean) as ResStreamMBS	441
* 21.5.17 QueryDiskGeometry(byref Cylinders as Int64, byref MediaType as Integer, byref TracksPerCylinder as Integer, byref SectorsPerTrack as Integer, byref BytesPerSector as Integer) as boolean	442
* 21.5.18 Read(count as Integer) as string	442
* 21.5.19 ReadBlock(count as Integer) as memoryblock	442
* 21.5.20 Readbyte as Integer	442
* 21.5.21 ReadLong as Integer	443
* 21.5.22 ReadShort as Integer	443
* 21.5.23 UnlockFileExclusive as boolean	443
* 21.5.24 WinCreateStream(file as folderitem, StreamName as String, WinShareMode as Integer = 0) as LargeBinaryStreamMBS	444
* 21.5.25 WinDeleteStream(file as folderitem, StreamName as String) as boolean	444
* 21.5.26 WinOpenStream(file as folderitem, StreamName as String, write as Boolean, WinShareMode as Integer = 0) as LargeBinaryStreamMBS	444
* 21.5.27 Write(data as string)	445
* 21.5.28 WriteBlock(data as memoryblock, count as Integer)	445
* 21.5.29 WriteByte(data as Integer)	445
* 21.5.30 WriteLong(data as Integer)	445
* 21.5.31 WriteShort(data as Integer)	446
* 21.5.33 CanWrite as boolean	446
* 21.5.34 EOF as boolean	446

* 21.5.35 Lasterror as Integer	446
* 21.5.36 Length as Int64	446
* 21.5.37 LittleEndian as boolean	447
* 21.5.38 Position as Int64	447
* 21.5.39 Yield as Boolean	447

	61
• 11 Controls	251
– 11.8.1 class Listbox	260
* 11.8.3 InvalidateCellThreadSafeMBS(Row as Integer, Column as Integer)	260

• 14 Currency, Date and Time Format	281
– ?? Globals	??
* 14.1.1 CDblMBS(text as string, byref value as Double, locale as string = "") as boolean	281
* 14.1.2 FormatDateMBS(format as string, value as date, locale as string = "") as string	282
* 14.1.3 FormatDateTimeMBS(format as string, value as dateTime, locale as string = "") as string	283
* 14.1.4 FormatMBS(format as string, value as Double, locale as string = "") as string	283
* 14.1.5 ParseDateMBS(format as string, text as string, byref value as date, locale as string = "") as boolean	284
* 14.1.6 ParseDateTimeMBS(format as string, text as string, byref value as dateTime, locale as string = "") as boolean	286
– 14.2.1 class LocaleMBS	286
* 14.2.3 Constructor	286
* 14.2.4 Locale(Locale as string = "") as LocaleMBS	286
* 14.2.6 CurrencySymbol as String	287
* 14.2.7 DecimalPoint as String	287
* 14.2.8 FracDigits as Integer	287
* 14.2.9 Grouping as String	288
* 14.2.10 IntCurrSymbol as String	288
* 14.2.11 IntFracDigits as Integer	288
* 14.2.12 IntNegCSPrecedes as Integer	288
* 14.2.13 IntNegSepBySpace as Integer	288
* 14.2.14 IntNegSignPosition as Integer	289
* 14.2.15 IntPosCSPrecedes as Integer	289
* 14.2.16 IntPosSepBySpace as Integer	289
* 14.2.17 IntPosSignPosition as Integer	289
* 14.2.18 monDecimalPoint as String	290
* 14.2.19 monGrouping as String	290
* 14.2.20 monThousandsSep as String	290
* 14.2.21 Name as String	290
* 14.2.22 NegativeSign as String	290
* 14.2.23 NegCSPrecedes as Boolean	291
* 14.2.24 NegSepBySpace as Boolean	291
* 14.2.25 NegSignPosition as Integer	291
* 14.2.26 PosCSPrecedes as Boolean	291
* 14.2.27 PositiveSign as String	291
* 14.2.28 PosSepBySpace as Boolean	292
* 14.2.29 PosSignPosition as Integer	292
* 14.2.30 ThousandsSep as String	292

• 44 XojoRuntime	1237
– ?? Globals	??
* 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()	1237
* 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()	1237
* 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()	1238
* 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()	1239
* 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean	1239
* 44.1.6 ArrayStringMBS(paramArray values as String) as String()	1240
* 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()	1240
* 44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer	1241
* 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
* 44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer	1242
* 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
* 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
* 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
* 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
* 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
* 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
* 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
* 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
* 44.1.18 GetTextMemoryAddressMBS(s as text) as integer	1246
* 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()	1247
* 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer	1248
* 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant	1248
* 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()	1249
* 44.1.23 GetVariantTypeMBS(va as variant) as Integer	1250
* 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
* 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)	1253
* 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)	1251

• 39 System	1143
– ?? Globals	??
* 39.1.6 AbortMBS	1146
* 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()	1147
* 39.1.1 CrashNiceMBS	1143
* 39.1.2 CrashUglyMBS	1143
* 39.1.3 DelayMBS(time as Double)	1143
* 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
* 39.1.7 ExitMBS(code as Integer)	1146
* 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
* 39.1.9 MillisecondsMBS as Double	1147
* 39.1.5 SleepMBS(time as Double)	1145

• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

* 38.1.37 GetStringsFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string()	1113
* 38.1.38 GetStringsFromDataMBS(data as String, MinLength as Integer = 0) as string()	1113
* 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string	1114
* 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean	1114
* 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean	1115
* 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string	1116
* 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer	1089
* 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer	1090
* 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer	1091
* 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer	1092
* 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer	1093
* 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer	1093
* 38.1.7 InStrBytesMBS(target as string, find as string) as Integer	1094
* 38.1.43 IsASCIIStringMBS(s as string) as boolean	1117
* 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean	1117
* 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double	1117
* 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double	1118
* 38.1.47 NativeStringMBS(s as string) as string	1119
* 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string	1119
* 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string	1095
* 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string	1120
* 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string	1120
* 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String	1121
* 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string	1121
* 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string	1122
* 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string()	1096
* 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string	1122
* 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer	1122
* 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer	1123
* 38.1.57 StringANDMBS(a as string,b as string) as string	1123

- * 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean 1124
- * 38.1.59 StringIsXMLreadyMBS(s as string) as boolean 1124
- * 38.1.60 StringORMBS(a as string,b as string) as string 1124
- * 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.63 StrMBS(d as Double) as string 1126
- * 38.1.64 UnicodeStringMBS(s as string) as string 1126

• 44 XojoRuntime	1237
– ?? Globals	??
* 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()	1237
* 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()	1237
* 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()	1238
* 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()	1239
* 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean	1239
* 44.1.6 ArrayStringMBS(paramArray values as String) as String()	1240
* 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()	1240
* 44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer	1241
* 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
* 44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer	1242
* 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
* 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
* 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
* 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
* 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
* 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
* 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
* 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
* 44.1.18 GetTextMemoryAddressMBS(s as text) as integer	1246
* 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()	1247
* 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer	1248
* 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant	1248
* 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()	1249
* 44.1.23 GetVariantTypeMBS(va as variant) as Integer	1250
* 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
* 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)	1253
* 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)	1251

	69
• 39 System	1143
– ?? Globals	??
* 39.1.6 AbortMBS	1146
* 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()	1147
* 39.1.1 CrashNiceMBS	1143
* 39.1.2 CrashUglyMBS	1143
* 39.1.3 DelayMBS(time as Double)	1143
* 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
* 39.1.7 ExitMBS(code as Integer)	1146
* 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
* 39.1.9 MillisecondsMBS as Double	1147
* 39.1.5 SleepMBS(time as Double)	1145

• 44 XojoRuntime	1237
– ?? Globals	??
* 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()	1237
* 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()	1237
* 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()	1238
* 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()	1239
* 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean	1239
* 44.1.6 ArrayStringMBS(paramArray values as String) as String()	1240
* 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()	1240
* 44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer	1241
* 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
* 44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer	1242
* 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
* 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
* 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
* 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
* 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
* 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
* 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
* 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
* 44.1.18 GetTextMemoryAddressMBS(s as text) as integer	1246
* 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()	1247
* 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer	1248
* 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant	1248
* 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()	1249
* 44.1.23 GetVariantTypeMBS(va as variant) as Integer	1250
* 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
* 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)	1253
* 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)	1251

	71
• 21 Files	379
– ?? Globals	??
* 21.2.2 AdminToolsMBS(domain as Integer) as folderitem	394
* 21.2.3 CookiesMBS as folderitem	395
* 21.2.1 ExchangeFilesMBS(first as folderitem, second as folderitem) as Integer	394
* 21.2.4 HistoryMBS as folderitem	395
* 21.2.5 InternetCacheMBS as folderitem	395
* 21.2.7 SetCurrentWorkingDirectoryMBS(path as folderitem) as boolean	395
* 21.2.8 SetDllDirectoryMBS(path as folderitem) as boolean	396
* 21.2.9 VolumeFreeSizePathMBS(Path as String) as Int64	397
* 21.2.10 VolumeSizePathMBS(Path as String) as Int64	397
* 21.2.6 WindowsStartMenuMBS(domain as Integer) as folderitem	395

• 44 XojoRuntime	1237
– ?? Globals	??
* 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()	1237
* 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()	1237
* 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()	1238
* 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()	1239
* 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean	1239
* 44.1.6 ArrayStringMBS(paramArray values as String) as String()	1240
* 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()	1240
* 44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer	1241
* 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
* 44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer	1242
* 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
* 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
* 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
* 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
* 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
* 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
* 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
* 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
* 44.1.18 GetTextMemoryAddressMBS(s as text) as integer	1246
* 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()	1247
* 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer	1248
* 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant	1248
* 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()	1249
* 44.1.23 GetVariantTypeMBS(va as variant) as Integer	1250
* 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
* 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)	1253
* 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)	1251

• 18 Encryption and Hash	345
– ?? Globals	??
* 18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32	353
* 18.1.3 CRC16MBS(data as string) as UInt16	346
* 18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	346
* 18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32	347
* 18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32	347
* 18.1.7 CRC_32OfStrMBS(s as String) as UInt32	347
* 18.1.8 CRC_CCITInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	348
* 18.1.9 CRC_CCITInMemMBS(address as Ptr, length as Integer) as UInt32	348
* 18.1.10 CRC_CCITTOfStrContMBS(s as String, prevCRC as UInt32) as UInt32	349
* 18.1.11 CRC_CCITTOfStrMBS(s as String) as UInt32	349
* 18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String	349
* 18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String	349
* 18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64	349
* 18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64	352
* 18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64	350
* 18.1.21 CurrentUnixTimeMBS as UInt64	353
* 18.1.22 DecodeFromBase32MBS(data as string) as String	353
* 18.1.23 EncodeToBase32MBS(data as string) as String	354
* 18.1.16 GetHash32MBS(s as string) as UInt32	350
* 18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16	351
* 18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string	351

• 21 Files	379
– ?? Globals	??
* 21.2.2 AdminToolsMBS(domain as Integer) as folderitem	394
* 21.2.3 CookiesMBS as folderitem	395
* 21.2.1 ExchangeFilesMBS(first as folderitem, second as folderitem) as Integer	394
* 21.2.4 HistoryMBS as folderitem	395
* 21.2.5 InternetCacheMBS as folderitem	395
* 21.2.7 SetCurrentWorkingDirectoryMBS(path as folderitem) as boolean	395
* 21.2.8 SetDllDirectoryMBS(path as folderitem) as boolean	396
* 21.2.9 VolumeFreeSizePathMBS(Path as String) as Int64	397
* 21.2.10 VolumeSizePathMBS(Path as String) as Int64	397
* 21.2.6 WindowsStartMenuMBS(domain as Integer) as folderitem	395

- 26 Math

535

- ?? Globals

??

- * 26.1.3 ACosHMBS(x as Double) as Double 536
 - * 26.1.4 ACosMBS(x as Double) as Double 536
 - * 26.1.5 ArithmeticShiftMBS(value as UInt64, count as Integer) as UInt64 537
 - * 26.1.6 ASinHMBS(x as Double) as Double 537
 - * 26.1.7 ASinMBS(x as Double) as Double 537
 - * 26.1.8 ATan2MBS(x as Double, y as Double) as Double 538
 - * 26.1.9 ATanHMBS(x as Double) as Double 538
 - * 26.1.10 ATanMBS(x as Double) as Double 539
 - * 26.1.11 BitClearMBS(value as UInt64, mask as UInt64) as UInt64 539
 - * 26.1.12 BitCountMBS(value as UInt64) as Integer 539
 - * 26.1.13 BitExclMBS(value as UInt64, bitNumber as Integer) as UInt64 540
 - * 26.1.14 BitInclMBS(value as UInt64, bitNumber as Integer) as UInt64 540
 - * 26.1.15 BitIsSetMBS(value as UInt64, bitNumber as Integer) as Boolean 540
 - * 26.1.16 BitValMBS(bitNumber as Integer) as UInt64 541
 - * 26.1.17 BitwiseDiffMBS(x as UInt64, y as UInt64) as UInt64 541
 - * 26.1.18 BitwiseNAndMBS(x as UInt64, y as UInt64) as UInt64 541
 - * 26.1.19 BitwiseNOrMBS(x as UInt64, y as UInt64) as UInt64 542
 - * 26.1.20 BitwiseNotMBS(value as UInt64) as UInt64 542
 - * 26.1.21 BitwiseRotateMBS(value as UInt64, count as Integer, offset as Integer, width as Integer) as UInt64 542
 - * 26.1.2 CompareNumbersMBS(v1 as Variant, v2 as Variant) as Integer 535
 - * 26.1.22 ConvertFromFloat16MBS(Number as UInt16) as Single 543
 - * 26.1.23 ConvertToFloat16MBS(Number as Single) as UInt16 543
 - * 26.1.24 CosHMBS(x as Double) as Double 544
 - * 26.1.25 CosMBS(x as Double) as Double 545
 - * 26.1.26 CurrencyAddMBS(value1 as Currency, value2 as Currency) as Currency 545
 - * 26.1.27 CurrencyDivMBS(value1 as Currency, value2 as Integer) as Currency 545
 - * 26.1.28 CurrencyMulMBS(value1 as Currency, value2 as Integer) as Currency 545
 - * 26.1.29 CurrencySubMBS(value1 as Currency, value2 as Currency) as Currency 546
 - * 26.1.30 CurrencyValueMBS(value as string) as Currency 546
 - * 26.1.31 DoubleToExtendedStrMBS(x as Double) as string 547
 - * 26.1.59 DoubleToInt64MBS(value as Double) as Int64 557
 - * 26.1.60 DoubleToUInt64MBS(value as Double) as UInt64 557
 - * 26.1.32 Exp2MBS(x as Double) as Double 547
 - * 26.1.33 ExpMBS(x as Double) as Double 547
 - * 26.1.34 ExtendedStrToDoubleMBS(v as string) as Double 548
 - * 26.1.35 FacMBS(x as Integer) as Double 548
 - * 26.1.36 FloorMBS(x as Double) as Double 549
 - * 26.1.37 FRExpMBS(inputx as Double, byref expValue as Integer) as Double 549

* 26.1.38 HiWordMBS(i as Integer) as Integer	550
* 26.1.39 HypotMBS(x as Double, y as Double) as Double	550
* 26.1.61 Int64ToDoubleMBS(value as Int64) as Double	558
* 26.1.40 IsFiniteMBS(x as Double) as boolean	550
* 26.1.41 IsInfMBS(x as Double) as boolean	550
* 26.1.42 IsNaNMBS(x as Double) as boolean	551
* 26.1.1 IsValidCreditCardNumberMBS(Number as String) as boolean	535
* 26.1.43 Log10MBS(x as Double) as Double	551
* 26.1.44 Log2MBS(x as Double) as Double	551
* 26.1.45 LogicalShiftMBS(value as UInt64, count as Integer) as UInt64	552
* 26.1.46 LogMBS(x as Double) as Double	552
* 26.1.47 LoWordMBS(i as Integer) as Integer	552
* 26.1.48 NormInvMBS(p as Double, mu as double = 0.0, sigma as double = 1.0) as double	553
* 26.1.49 PowMBS(x as Double, y as Double) as Double	553
* 26.1.50 RandomExponentialDistributionMBS(lambda as Double) as double	554
* 26.1.51 RandomNormalDistributionMBS(Mean as Double, StdDev as Double) as double	554
* 26.1.52 RandomPoissonDistributionMBS(Mean as Integer) as Integer	554
* 26.1.53 RoundMBS(x as Double, decimals as Integer = 0) as Double	555
* 26.1.54 SinHMBS(x as Double) as Double	555
* 26.1.55 SinMBS(x as Double) as Double	556
* 26.1.56 SqrtMBS(x as Double, y as Double) as Double	556
* 26.1.57 TanHMBS(x as Double) as Double	556
* 26.1.58 TanMBS(x as Double) as Double	557
* 26.1.62 UInt64ToDoubleMBS(value as UInt64) as Double	558

	77
• 27 MemoryBlock	563
– ?? Globals	??
* 27.1.3 Memoryblock2ptrMBS(mem as memoryblock) as Integer	563
* 27.1.2 NewMemoryBlockFromPtrMBS(ptr as Integer) as memoryblock	563
* 27.1.1 NewMemoryBlockWithBytesMBS(Data as Ptr, size as Integer) as memoryblock	563
* 27.1.4 ptr2MemoryblockMBS(Value as Integer) as memoryblock	564
– 27.2.1 class Memoryblock	564
* 27.2.3 AddressMBS(offset as Int64 = 0) as UInt64	564
* 27.2.4 AddressPtrMBS(offset as Int64 = 0) as Ptr	564
* 27.2.5 AndBitsMBS(Second as memoryblock, Dest as memoryblock=nil) as memoryblock	564
* 27.2.6 AndBitsMBS(Second as memoryblock, Mask as Integer, Dest as memoryblock=nil) as memoryblock	565
* 27.2.7 AppendMBS(other as memoryblock) as memoryblock	566
* 27.2.8 BitwiseAndMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock	566
* 27.2.9 BitwiseOrMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock	568
* 27.2.10 BitwiseXOrMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock	569
* 27.2.11 BytesEqualMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer) as Boolean	570
* 27.2.12 BytesZeroMBS(srcOfs as Integer, numBytes as Integer) as Boolean	571
* 27.2.13 ClearBitMBS(Bit as UInt64)	572
* 27.2.14 ConvertRGB12BitTo8BitMBS(Width as Integer)	572
* 27.2.15 CopyBytesFromMacHandleMBS(srcHandle as Integer, numBytes as Integer, destOfs as Integer)	572
* 27.2.16 CopyBytesFromMacPtrMBS(srcPtr as Ptr, numBytes as Integer, destOfs as Integer)	573
* 27.2.17 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer)	573
* 27.2.18 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destOfs as Integer)	574
* 27.2.19 CopyBytesToMacHandleMBS(srcOfs as Integer, numBytes as Integer, destHandle as Integer)	574
* 27.2.20 CopyBytesToMacPtrMBS(srcOfs as Integer, numBytes as Integer, destPtr as Ptr)	574
* 27.2.21 CopyByteToUShortMBS(dest as memoryblock, SourceOffset as Integer, DestinationOffset as Integer, ByteCount as Integer, divisor as Integer)	574
* 27.2.22 CopyNthBitsMBS(source as memoryblock, SourceOffsetBits as Integer, DestinationOffsetBits as Integer, BitCount as Integer, StepCount as Integer, NumberOfSteps as Integer) as boolean	575
* 27.2.23 CopyNthBytesMBS(source as memoryblock, SourceOffsetBytes as Integer, DestinationOffsetBytes as Integer, ByteCount as Integer, StepCount as Integer, NumberOfSteps as Integer) as boolean	575

* 27.2.24 CopyUShortToByteMBS(dest as memoryblock, SourceOffset as Integer, DestinationOffset as Integer, ByteCount as Integer, divisor as Integer)	575
* 27.2.56 ExpandBitsMBS(dest as memoryblock, SourceByteCount as Integer, LowValue as Integer = 0, HighValue as Integer = 255) as boolean	589
* 27.2.57 ExtractBitsMBS(Mask as Integer, Dest as memoryblock=nil) as memoryblock	590
* 27.2.58 FillBytesMBS(offset as Integer, count as Integer, value as Integer)	590
* 27.2.59 FindByteMBS(srcOfs as Integer, numBytes as Integer, byteValue as Integer) as Integer	590
* 27.2.60 FindByteMBS(values() as UInt8, StartByteOffset as Integer = 0) as Integer	591
* 27.2.61 FindBytesMBS(srcOfs as Integer, maxBytes as Integer, target as memoryBlock, targOfs as Integer, targLen as Integer) as Integer	592
* 27.2.62 FindNotByteMBS(srcOfs as Integer, numBytes as Integer, byteValue as Integer) as Integer	592
* 27.2.63 FindStringMBS(srcOfs as Integer, maxBytes as Integer, target as String) as Integer	593
* 27.2.64 GetBitMBS(Bit as UInt64) as Integer	593
* 27.2.65 GetStringMBS(offset as Integer, numBytes as Integer) as String	594
* 27.2.66 InvertBytesMBS(offset as Integer, count as Integer)	594
* 27.2.67 IsBitSetMBS(Bit as UInt64) as Boolean	594
* 27.2.68 LeftMBS(length as Integer) as memoryblock	595
* 27.2.69 MaxMBS(firstMem as Ptr, secondMem as Ptr, BitSize as Integer = 8, Signed as Boolean = false, offsetByte as Integer = 0, lengthBytes as Integer = 0) as boolean	595
* 27.2.70 MidMBS(offset as Integer) as memoryblock	596
* 27.2.71 MidMBS(offset as Integer, length as Integer) as memoryblock	596
* 27.2.72 MinMBS(firstMem as Ptr, secondMem as Ptr, BitSize as Integer = 8, Signed as Boolean = false, offsetByte as Integer = 0, lengthBytes as Integer = 0) as boolean	596
* 27.2.73 MirrorBitsInBytesMBS(offsetByte as Integer, lengthByte as Integer)	597
* 27.2.74 MirrorBitsMBS(offsetBit as Integer, lengthBit as Integer)	598
* 27.2.75 MirrorBytesMBS(offsetByte as Integer = 0, lengthByte as Integer = -1)	598
* 27.2.76 MultiplyUInt16MBS(Factor as Double, offsetByte as Integer = 0, lengthBytes as Integer = 0, MultiplyOnlyOver as Integer = -1)	598
* 27.2.77 MultiplyUInt8MBS(Factor as Double, offsetByte as Integer = 0, lengthBytes as Integer = 0, MultiplyOnlyOver as Integer = -1)	599
* 27.2.78 RightMBS(length as Integer) as memoryblock	599
* 27.2.79 SetBitMBS(Bit as UInt64)	600
* 27.2.80 SetBitMBS(Bit as UInt64, Value as Boolean)	600
* 27.2.81 SetBitMBS(Bit as UInt64, Value as Integer)	601
* 27.2.82 SetStringMBS(str as String, offset as Integer)	601
* 27.2.83 SwapBytes16MBS(offset as Integer, numBytes as Integer)	601
* 27.2.84 SwapBytes32MBS(offset as Integer, numBytes as Integer)	602
* 27.2.85 SwapBytesMBS(offset as Integer, numBytes as Integer)	603
* 27.2.87 OSTypeMBS(offset as Integer) as String	604

• 18 Encryption and Hash	345
– 27.2.1 class Memoryblock	564
* 27.2.25 CRC_32ContMBS(offset as Integer, numBytes as Integer, prevCRC as UInt32) as UInt32	576
* 27.2.26 CRC_32MBS(offset as Integer, numBytes as Integer) as UInt32	576
* 27.2.27 CRC_CCITTContMBS(offset as Integer, numBytes as Integer, prevCRC as UInt32) as UInt32	576
* 27.2.28 CRC_CCITTMBS(offset as Integer, numBytes as Integer) as UInt32	576
* 27.2.29 CRC_DillonMBS(bitWidth as Integer, offset as Integer, numBytes as Integer) as String	576
– ?? Globals	??
* 18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16	345
* 18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32	353
* 18.1.3 CRC16MBS(data as string) as UInt16	346
* 18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	346
* 18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32	347
* 18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32	347
* 18.1.7 CRC_32OfStrMBS(s as String) as UInt32	347
* 18.1.8 CRC_CCITTCInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32	348
* 18.1.9 CRC_CCITTCInMemMBS(address as Ptr, length as Integer) as UInt32	348
* 18.1.10 CRC_CCITTCOfStrContMBS(s as String, prevCRC as UInt32) as UInt32	349
* 18.1.11 CRC_CCITTCOfStrMBS(s as String) as UInt32	349
* 18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String	349
* 18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String	349
* 18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64	349
* 18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64	352
* 18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64	350
* 18.1.21 CurrentUnixTimeMBS as UInt64	353
* 18.1.22 DecodeFromBase32MBS(data as string) as String	353
* 18.1.23 EncodeToBase32MBS(data as string) as String	354
* 18.1.16 GetHash32MBS(s as string) as UInt32	350
* 18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16	351
* 18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string	351

• 19 Endian	359
– 27.2.1 class Memoryblock	564
* 27.2.30 EndianS16_BtoLMBS(offset as Integer,count as Integer)	577
* 27.2.31 EndianS16_BtoNMBS(offset as Integer,count as Integer)	577
* 27.2.32 EndianS16_LtoBMBS(offset as Integer,count as Integer)	578
* 27.2.33 EndianS16_LtoNMBS(offset as Integer,count as Integer)	578
* 27.2.34 EndianS16_NtoBMBS(offset as Integer,count as Integer)	579
* 27.2.35 EndianS16_NtoLMBS(offset as Integer,count as Integer)	579
* 27.2.36 EndianS32_BtoLMBS(offset as Integer,count as Integer)	580
* 27.2.37 EndianS32_BtoNMBS(offset as Integer,count as Integer)	580
* 27.2.38 EndianS32_LtoBMBS(offset as Integer,count as Integer)	581
* 27.2.39 EndianS32_LtoNMBS(offset as Integer,count as Integer)	581
* 27.2.40 EndianS32_NtoBMBS(offset as Integer,count as Integer)	582
* 27.2.41 EndianS32_NtoLMBS(offset as Integer,count as Integer)	582
* 27.2.42 EndianSwap16MBS(offset as Integer,count as Integer)	583
* 27.2.43 EndianSwap32MBS(offset as Integer,count as Integer)	583
* 27.2.44 EndianU16_BtoLMBS(offset as Integer,count as Integer)	583
* 27.2.45 EndianU16_BtoNMBS(offset as Integer,count as Integer)	584
* 27.2.46 EndianU16_LtoBMBS(offset as Integer,count as Integer)	584
* 27.2.47 EndianU16_LtoNMBS(offset as Integer,count as Integer)	585
* 27.2.48 EndianU16_NtoBMBS(offset as Integer,count as Integer)	585
* 27.2.49 EndianU16_NtoLMBS(offset as Integer,count as Integer)	586
* 27.2.50 EndianU32_BtoLMBS(offset as Integer,count as Integer)	586
* 27.2.51 EndianU32_BtoNMBS(offset as Integer,count as Integer)	587
* 27.2.52 EndianU32_LtoBMBS(offset as Integer,count as Integer)	587
* 27.2.53 EndianU32_LtoNMBS(offset as Integer,count as Integer)	588
* 27.2.54 EndianU32_NtoBMBS(offset as Integer,count as Integer)	588
* 27.2.55 EndianU32_NtoLMBS(offset as Integer,count as Integer)	589

	81
• 27 MemoryBlock	563
– 27.2.1 class Memoryblock	564
* 27.2.3 AddressMBS(offset as Int64 = 0) as UInt64	564
* 27.2.4 AddressPtrMBS(offset as Int64 = 0) as Ptr	564
* 27.2.5 AndBitsMBS(Second as memoryblock, Dest as memoryblock=nil) as memoryblock	564
* 27.2.6 AndBitsMBS(Second as memoryblock, Mask as Integer, Dest as memoryblock=nil) as memoryblock	565
* 27.2.7 AppendMBS(other as memoryblock) as memoryblock	566
* 27.2.8 BitwiseAndMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock	566
* 27.2.9 BitwiseOrMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock	568
* 27.2.10 BitwiseXOrMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock	569
* 27.2.11 BytesEqualMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer) as Boolean	570
* 27.2.12 BytesZeroMBS(srcOfs as Integer, numBytes as Integer) as Boolean	571
* 27.2.13 ClearBitMBS(Bit as UInt64)	572
* 27.2.14 ConvertRGB12BitTo8BitMBS(Width as Integer)	572
* 27.2.15 CopyBytesFromMacHandleMBS(srcHandle as Integer, numBytes as Integer, destOfs as Integer)	572
* 27.2.16 CopyBytesFromMacPtrMBS(srcPtr as Ptr, numBytes as Integer, destOfs as Integer)	573
* 27.2.17 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer)	573
* 27.2.18 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destOfs as Integer)	574
* 27.2.19 CopyBytesToMacHandleMBS(srcOfs as Integer, numBytes as Integer, destHandle as Integer)	574
* 27.2.20 CopyBytesToMacPtrMBS(srcOfs as Integer, numBytes as Integer, destPtr as Ptr)	574
* 27.2.21 CopyByteToUShortMBS(dest as memoryblock, SourceOffset as Integer, DestinationOffset as Integer, ByteCount as Integer, divisor as Integer)	574
* 27.2.22 CopyNthBitsMBS(source as memoryblock, SourceOffsetBits as Integer, DestinationOffsetBits as Integer, BitCount as Integer, StepCount as Integer, NumberOfSteps as Integer) as boolean	575
* 27.2.23 CopyNthBytesMBS(source as memoryblock, SourceOffsetBytes as Integer, DestinationOffsetBytes as Integer, ByteCount as Integer, StepCount as Integer, NumberOfSteps as Integer) as boolean	575
* 27.2.24 CopyUShortToByteMBS(dest as memoryblock, SourceOffset as Integer, DestinationOffset as Integer, ByteCount as Integer, divisor as Integer)	575
* 27.2.56 ExpandBitsMBS(dest as memoryblock, SourceByteCount as Integer, LowValue as Integer = 0, HighValue as Integer = 255) as boolean	589
* 27.2.57 ExtractBitsMBS(Mask as Integer, Dest as memoryblock=nil) as memoryblock	590

* 27.2.58	FillBytesMBS(offset as Integer, count as Integer, value as Integer)	590
* 27.2.59	FindByteMBS(srcOfs as Integer, numBytes as Integer, byteValue as Integer) as Integer	590
* 27.2.60	FindByteMBS(values() as UInt8, StartByteOffset as Integer = 0) as Integer	591
* 27.2.61	FindBytesMBS(srcOfs as Integer, maxBytes as Integer, target as memoryBlock, targOfs as Integer, targLen as Integer) as Integer	592
* 27.2.62	FindNotByteMBS(srcOfs as Integer, numBytes as Integer, byteValue as Integer) as Integer	592
* 27.2.63	FindStringMBS(srcOfs as Integer, maxBytes as Integer, target as String) as Integer	593
* 27.2.64	GetBitMBS(Bit as UInt64) as Integer	593
* 27.2.65	GetStringMBS(offset as Integer, numBytes as Integer) as String	594
* 27.2.66	InvertBytesMBS(offset as Integer, count as Integer)	594
* 27.2.67	IsBitSetMBS(Bit as UInt64) as Boolean	594
* 27.2.68	LeftMBS(length as Integer) as memoryblock	595
* 27.2.69	MaxMBS(firstMem as Ptr, secondMem as Ptr, BitSize as Integer = 8, Signed as Boolean = false, offsetByte as Integer = 0, lengthBytes as Integer = 0) as boolean	595
* 27.2.70	MidMBS(offset as Integer) as memoryblock	596
* 27.2.71	MidMBS(offset as Integer, length as Integer) as memoryblock	596
* 27.2.72	MinMBS(firstMem as Ptr, secondMem as Ptr, BitSize as Integer = 8, Signed as Boolean = false, offsetByte as Integer = 0, lengthBytes as Integer = 0) as boolean	596
* 27.2.73	MirrorBitsInBytesMBS(offsetByte as Integer, lengthByte as Integer)	597
* 27.2.74	MirrorBitsMBS(offsetBit as Integer, lengthBit as Integer)	598
* 27.2.75	MirrorBytesMBS(offsetByte as Integer = 0, lengthByte as Integer = -1)	598
* 27.2.76	MultiplyUInt16MBS(Factor as Double, offsetByte as Integer = 0, lengthBytes as Integer = 0, MultiplyOnlyOver as Integer = -1)	598
* 27.2.77	MultiplyUInt8MBS(Factor as Double, offsetByte as Integer = 0, lengthBytes as Integer = 0, MultiplyOnlyOver as Integer = -1)	599
* 27.2.78	RightMBS(length as Integer) as memoryblock	599
* 27.2.79	SetBitMBS(Bit as UInt64)	600
* 27.2.80	SetBitMBS(Bit as UInt64, Value as Boolean)	600
* 27.2.81	SetBitMBS(Bit as UInt64, Value as Integer)	601
* 27.2.82	SetStringMBS(str as String, offset as Integer)	601
* 27.2.83	SwapBytes16MBS(offset as Integer, numBytes as Integer)	601
* 27.2.84	SwapBytes32MBS(offset as Integer, numBytes as Integer)	602
* 27.2.85	SwapBytesMBS(offset as Integer, numBytes as Integer)	603
* 27.2.87	OSTypeMBS(offset as Integer) as String	604
- ??	Globals	??
* 27.1.3	Memoryblock2ptrMBS(mem as memoryblock) as Integer	563
* 27.1.2	NewMemoryBlockFromPtrMBS(ptr as Integer) as memoryblock	563
* 27.1.1	NewMemoryBlockWithBytesMBS(Data as Ptr, size as Integer) as memoryblock	563
* 27.1.4	ptr2MemoryblockMBS(Value as Integer) as memoryblock	564

	83
– 27.3.1 class MemoryBlockMBS	605
* 27.3.3 Close	605
* 27.3.4 Constructor	606
* 27.3.5 Constructor(Mem as MemoryBlock)	606
* 27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0)	607
* 27.3.7 Constructor(Size as Int64)	607
* 27.3.8 Constructor(Str as String)	607
* 27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0)	608
* 27.3.10 Create(size as Int64) as boolean	609
* 27.3.11 Resize(Size as Int64) as boolean	609
* 27.3.13 Address as Int64	610
* 27.3.14 Memory as Memoryblock	610
* 27.3.15 Size as Int64	610
– 27.4.1 class MemoryStorageMBS	611
* 27.4.3 Constructor(Size as Int64 = 0)	611
* 27.4.4 Destructor	611
* 27.4.5 MemoryValue(Offset as Int64, Assigns s as MemoryBlock)	612
* 27.4.6 MemoryValue(Offset as Int64, Size as Int64) as MemoryBlock	612
* 27.4.7 StringValue(Offset as Int64, Assigns s as String)	612
* 27.4.8 StringValue(Offset as Int64, Size as Int64) as String	612
* 27.4.10 MemoryValue as MemoryBlock	613
* 27.4.11 Size as Integer	613
* 27.4.12 SizeAllocated as Integer	613
* 27.4.13 StringValue as String	613

- **11 Controls** 251
 - **?? Globals** ??
 - * 11.1.2 ShowModalThreadSafeMBS(extends theMessageDialog as MessageDialog) 251
 - * 11.1.1 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as DesktopWindow) 251
 - * 11.1.3 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as window) 252

	85
• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock,len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock,len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

• 29 Process	631
– 29.9.1 class MutexMBS	664
* 29.9.3 Lock	664
* 29.9.4 TryLock as boolean	664
* 29.9.5 Unlock	665
* 29.9.7 Handle as Integer	665
* 29.9.8 Tag as Variant	666
– 29.10.1 class NamedMutexMBS	667
* 29.10.3 Constructor(Name as string)	667
* 29.10.4 Lock	667
* 29.10.5 TryLock as boolean	668
* 29.10.6 Unlock	668
* 29.10.8 Handle as Integer	668
* 29.10.9 Locked as Boolean	668
* 29.10.10 Name as String	668
* 29.10.11 Tag as Variant	669

	87
• 28 Navigation	615
– 28.1.1 class OpenFileDialogFileTypeMBS	615
* 28.1.3 Close	615
* 28.1.5 Extension as String	615
* 28.1.6 Name as String	616
* 28.1.7 Type as String	616
– 28.2.1 class OpenFileDialogItemMBS	617
* 28.2.3 ReadData as String	617
* 28.2.5 Handle as Integer	617
* 28.2.6 Name as String	617
* 28.2.7 Path as String	618
* 28.2.8 URL as String	618
– 28.3.1 class OpenFileDialogMBS	619
* 28.3.3 AddType(t as OpenFileDialogFileTypeMBS)	620
* 28.3.4 ClearTypes	620
* 28.3.5 CountTypes as Integer	620
* 28.3.6 Files as FolderItem()	620
* 28.3.7 Files(index as Integer) as folderitem	621
* 28.3.8 GetCustomImageHeight as Integer	621
* 28.3.9 GetType(index as Integer) as OpenFileDialogFileTypeMBS	621
* 28.3.10 Items as OpenFileDialogItemMBS()	621
* 28.3.11 RefreshCustomImage	622
* 28.3.12 ShowDialog	622
* 28.3.14 accessoryView as Variant	622
* 28.3.15 ActionButtonLabel as String	622
* 28.3.16 AllowFolderSelection as Boolean	623
* 28.3.17 CancelButtonLabel as String	624
* 28.3.18 ClientName as String	624
* 28.3.19 Creator as String	624
* 28.3.20 CustomPicture as Picture	625
* 28.3.21 File as FolderItem	625
* 28.3.22 FileCount as Integer	625
* 28.3.23 InitialDirectory as FolderItem	626
* 28.3.24 Lasterror as Integer	626
* 28.3.25 Left as Integer	626
* 28.3.26 MultipleSelection as Boolean	626
* 28.3.27 ParentWindow as Variant	627
* 28.3.28 PromptText as String	627
* 28.3.29 ResolveAliases as Boolean	627
* 28.3.30 ShowHiddenFiles as Boolean	628
* 28.3.31 Top as Integer	628

* 28.3.32 TreatFilePackagesAsDirectories as Boolean	628
* 28.3.33 UseCustomPicture as Integer	628
* 28.3.34 WindowTitle as String	629
* 28.3.36 FilterItem(file as folderitem, filterMode as Integer) as boolean	629
* 28.3.37 SelectionChanged(file as folderitem)	629

• 29 Process	631
– ?? Globals	??
* 29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean	643
* 29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean	643
* 29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean	644
* 29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean	645
* 29.4.5 CallMethodMBS(target as object, name as string) as boolean	646
* 29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean	646
* 29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean	647
* 29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean	647
* 29.4.9 CallMethodOnMainThreadMBS(target as object, name as string) as boolean	648
* 29.4.10 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant) as boolean	649
* 29.4.11 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean	649
* 29.4.12 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean	650
* 29.4.14 CountProcessesMBS as Integer	651
* 29.4.13 SetThreadNameMBS(name as string)	651
– 29.11.1 class ProcessMBS	670
* 29.11.3 Bundle as folderitem	671
* 29.11.4 BundleID as string	671
* 29.11.5 CurrentProcessID as Integer	671
* 29.11.6 GetCurrentProcess	671
* 29.11.7 GetfirstProcess	672
* 29.11.8 GetFrontProcess	672
* 29.11.9 GetNextProcess as boolean	672
* 29.11.10 GetProcess(serial as memoryblock)	672
* 29.11.11 KillProcess as Integer	673
* 29.11.12 KillProcess(ProcessID as Integer, timeout as Integer) as Integer	673
* 29.11.13 KillProcess(timeoutms as Integer) as Integer	674
* 29.11.14 MacProcessSerial as memoryblock	674
* 29.11.15 ProcessInformationCFDictionary as object	675
* 29.11.16 QuitProcess as Integer	675
* 29.11.17 SameAs(other as ProcessMBS) as boolean	676
* 29.11.18 SetFrontProcessWithOptions(options as Integer)	676

* 29.11.19 SetServiceMode(ismode as boolean) as boolean	677
* 29.11.20 TransformProcessType(mode as Integer) as Integer	677
* 29.11.21 TransformToForegroundApplication as Integer	677
* 29.11.22 Update	678
* 29.11.23 WinFullProcessImagePath as string	678
* 29.11.24 WinGetPriorityClass(ProcessID as Integer) as Integer	678
* 29.11.25 WinModulePath as string	679
* 29.11.26 WinProcessImagePath as string	679
* 29.11.27 WinSetPriorityClass(ProcessID as Integer, PriorityClass as Integer) as Integer	680
* 29.11.29 CPUTime as Integer	681
* 29.11.30 CurrentProcess as boolean	681
* 29.11.31 flags as Integer	682
* 29.11.32 FrontProcess as boolean	683
* 29.11.33 lasterror as Integer	684
* 29.11.34 LaunchProcess as ProcessMBS	684
* 29.11.35 MacCreator as string	684
* 29.11.36 MacType as string	684
* 29.11.37 MemoryFree as Integer	685
* 29.11.38 MemorySize as Integer	685
* 29.11.39 Name as string	685
* 29.11.40 Path as folderitem	685
* 29.11.41 ProcessID as Integer	686
* 29.11.42 Visible as boolean	686
* 29.11.43 Priority as Integer	687
* 29.11.44 Priority(ProcessID as Integer) as Integer	687

	91
• 11 Controls	251
– 11.9.1 class ProgressBar	261
* 11.9.3 SetMaximumThreadSafeMBS(maximum as Integer)	261
* 11.9.4 SetMinimumThreadSafeMBS(minimum as Integer)	261
* 11.9.5 SetValueThreadSafeMBS(value as Integer)	262

• 30 RAMStream	689
– 30.1.1 class RAMStreamMBS	689
* 30.1.3 close	689
* 30.1.4 Constructor(InitialSize as Integer=0)	690
* 30.1.5 Look(count as Integer) as string	690
* 30.1.6 LookBlock(count as Integer) as memoryblock	690
* 30.1.7 LookByte as Integer	691
* 30.1.8 LookLong as Integer	691
* 30.1.9 LookShort as Integer	691
* 30.1.10 Read(count as Integer) as string	691
* 30.1.11 ReadBlock(count as Integer) as memoryblock	692
* 30.1.12 Readbyte as Integer	692
* 30.1.13 ReadLong as Integer	692
* 30.1.14 ReadShort as Integer	692
* 30.1.15 Write(data as string)	693
* 30.1.16 WriteBlock(data as memoryblock,count as Integer)	693
* 30.1.17 WriteByte(data as Integer)	693
* 30.1.18 WriteLong(data as Integer)	693
* 30.1.19 WriteShort(data as Integer)	694
* 30.1.21 EOF as boolean	694
* 30.1.22 GrowSize as Integer	694
* 30.1.23 Length as Integer	694
* 30.1.24 LittleEndian as boolean	694
* 30.1.25 MemoryUsed as Integer	695
* 30.1.26 Position as Integer	695
– ?? Globals	??
* 30.2.1 CreateRamStreamMBS(InitialSize as Integer = 0) as RamStreamMBS	696

	93
• 11 Controls	251
– 11.10.1 class RectControl	263
* 11.10.3 InvalidateThreadSafeMBS(EraseBackground as boolean = true)	263
* 11.10.4 InvalidateThreadSafeMBS(X as Integer, Y as Integer, Width as Integer, Height as Integer, EraseBackground as boolean = true)	263
* 11.10.5 RefreshThreadSafeMBS(EraseBackground as boolean = true)	264
* 11.10.6 SetEnabledThreadSafeMBS(value as boolean)	264
* 11.10.7 SetVisibleThreadSafeMBS(value as boolean)	264

• 31 Registration	697
– 31.1.1 class RegistrationEngineMBS	697
* 31.1.3 Calc as string	698
* 31.1.4 Verify(s as string) as boolean	698
* 31.1.6 Alphabet as String	699
* 31.1.7 BlockLength as Integer	699
* 31.1.8 Delimiter as String	700
* 31.1.9 Mode as Integer	700
* 31.1.10 NumberLength as Integer	700
* 31.1.11 Platform as Integer	701
* 31.1.12 PlatformSpecificKeys as Boolean	701
* 31.1.13 Prefix as String	701
* 31.1.14 Seed as Integer	702
* 31.1.15 Suffix as String	702
* 31.1.16 Field(index as Integer) as string	702

• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

* 38.1.37 GetStringsFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string()	1113
* 38.1.38 GetStringsFromDataMBS(data as String, MinLength as Integer = 0) as string()	1113
* 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string	1114
* 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean	1114
* 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean	1115
* 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string	1116
* 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer	1089
* 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer	1090
* 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer	1091
* 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer	1092
* 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer	1093
* 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer	1093
* 38.1.7 InStrBytesMBS(target as string, find as string) as Integer	1094
* 38.1.43 IsASCIIStringMBS(s as string) as boolean	1117
* 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean	1117
* 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double	1117
* 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double	1118
* 38.1.47 NativeStringMBS(s as string) as string	1119
* 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string	1119
* 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string	1095
* 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string	1120
* 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string	1120
* 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String	1121
* 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string	1121
* 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string	1122
* 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string()	1096
* 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string	1122
* 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer	1122
* 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer	1123
* 38.1.57 StringANDMBS(a as string,b as string) as string	1123

- * 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean 1124
- * 38.1.59 StringIsXMLreadyMBS(s as string) as boolean 1124
- * 38.1.60 StringORMBS(a as string,b as string) as string 1124
- * 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.63 StrMBS(d as Double) as string 1126
- * 38.1.64 UnicodeStringMBS(s as string) as string 1126

• 32 Resolution	703
– 32.3.1 class ResolutionMBS	712
* 32.3.3 Switch As Boolean	712
* 32.3.5 Depth as Integer	712
* 32.3.6 displaynum as Integer	713
* 32.3.7 Height as Integer	713
* 32.3.8 hz as Integer	713
* 32.3.9 issafe as boolean	713
* 32.3.10 Left as Integer	713
* 32.3.11 ResolutionNum as Integer	714
* 32.3.12 Top as Integer	714
* 32.3.13 Width as Integer	714
– ?? Globals	??
* 32.2.1 DisplayCountMBS as Integer	711
* 32.2.2 GetDisplayMBS(num as Integer) As DisplayMBS	711
* 32.2.3 ResolutionLibraryPresentMBS as boolean	711
* 32.2.4 UpdateDisplayCountMBS	712

	99
• 33 SerialPort	715
– 33.1.1 class SerialPortMBS	715
* 33.1.3 AvailableBytes as Integer	716
* 33.1.4 Clear	716
* 33.1.5 Close	716
* 33.1.6 Constructor	716
* 33.1.7 HasDataAvailable as Boolean	716
* 33.1.8 HasLine as Boolean	716
* 33.1.9 List(Mode as Integer = 0) as String()	717
* 33.1.10 Open(Name as String, Index as Integer = 0)	717
* 33.1.11 OpenPath(Path as String)	718
* 33.1.12 Poll	718
* 33.1.13 Read(MaxByteSizeToRead as Integer) as String	718
* 33.1.14 ReadByte as Integer	718
* 33.1.15 ReadLine as String	718
* 33.1.16 Resume	719
* 33.1.17 Suspend	719
* 33.1.18 Write(value as String) as Integer	719
* 33.1.19 WriteByte(value as Integer) as Integer	719
* 33.1.21 Handle as Integer	719
* 33.1.22 Lasterror as Integer	720
* 33.1.23 Path as String	720
* 33.1.24 Suspended as Boolean	720
* 33.1.25 Tag as Variant	720
* 33.1.26 BaudRate as Integer	721
* 33.1.27 CTS as Boolean	721
* 33.1.28 DataBits as Integer	721
* 33.1.29 DSR as Boolean	721
* 33.1.30 DTR as Boolean	721
* 33.1.31 Parity as Integer	722
* 33.1.32 RTS as Boolean	722
* 33.1.33 StopBits as Double	722
* 33.1.34 WinRTSControl as Integer	722
* 33.1.35 XON as Boolean	723
* 33.1.37 DataAvailable	723

• 34 Shell	725
– 34.2.1 class ShellMBS	727
* 34.2.3 Arguments as String()	728
* 34.2.4 CloseInput	728
* 34.2.5 CloseStreams	729
* 34.2.6 Constructor	729
* 34.2.7 Destructor	729
* 34.2.8 Execute(Executable as String)	729
* 34.2.9 ExitCode as Integer	730
* 34.2.10 Poll	730
* 34.2.11 ReadError as String	730
* 34.2.12 ReadOutput as String	730
* 34.2.13 SetArguments(arguments() as String)	730
* 34.2.14 Terminate(WinExitCode as Integer = 255) as boolean	731
* 34.2.15 Wait(TimeOut as double = 30.0)	731
* 34.2.16 WriteInput(data as string) as Integer	731
* 34.2.18 ApplicationName as String	731
* 34.2.19 AvailableBytesError as Integer	731
* 34.2.20 AvailableBytesOutput as Integer	732
* 34.2.21 CurrentDirectory as String	732
* 34.2.22 Domain as String	732
* 34.2.23 Environment as Dictionary	732
* 34.2.24 ErrorCode as Integer	732
* 34.2.25 Executable as String	733
* 34.2.26 IsRunning as Boolean	733
* 34.2.27 MergeErrorAndOutput as Boolean	733
* 34.2.28 Password as String	733
* 34.2.29 PID as Integer	734
* 34.2.30 ProcessHandle as Integer	734
* 34.2.31 Tag as Variant	734
* 34.2.32 ThreadHandle as Integer	734
* 34.2.33 Username as String	734
* 34.2.35 Completed	735
* 34.2.36 DataAvailable(AvailableBytesError as Integer, AvailableBytesOutput as Integer)	735

	101
• 39 System	1143
– 39.2.1 class SignalHandlerMBS	1148
* 39.2.3 alarm(seconds as Integer)	1149
* 39.2.4 ClearFlag(signalIndex as Integer)	1149
* 39.2.5 ClearFlags	1149
* 39.2.6 ClearStacktrace(signalIndex as Integer)	1149
* 39.2.7 Close	1150
* 39.2.8 IsFlagSet(signalIndex as Integer) as boolean	1150
* 39.2.9 QueryStacktrace(signalIndex as Integer, skip as Integer = 2) as string()	1150
* 39.2.10 SendSignal(PID as Integer, Signal as Integer) as boolean	1150
* 39.2.11 SendSignalToSelf(Signal as Integer) as boolean	1150
* 39.2.12 SetDefaultHandler(signalIndex as Integer) as boolean	1151
* 39.2.13 SetEventHandler(signalIndex as Integer, CollectStackTrace as boolean = false) as boolean	1151
* 39.2.14 SetFlagHandler(signalIndex as Integer, CollectStackTrace as boolean = false) as boolean	1151
* 39.2.15 SetIgnore(signalIndex as Integer) as boolean	1152
* 39.2.16 SetPrintBacktraceAndAbortHandler(signalIndex as integer) as boolean	1152
* 39.2.17 SignalStatus(signalIndex as Integer) as Integer	1152
* 39.2.19 Signal(n as Integer)	1153

• 15 Declare	299
– 15.2.1 class SoftDeclareMBS	306
* 15.2.3 CallFunction(param as string,data as memoryblock) as boolean	307
* 15.2.4 CallFunction(paramcount as Integer,data as memoryblock) as boolean	308
* 15.2.5 CallFunctionDouble(param as string,data as memoryblock) as boolean	310
* 15.2.6 CallFunctionDouble(paramcount as Integer,data as memoryblock) as boolean	310
* 15.2.7 CallFunctionInteger64(param as string,data as memoryblock) as boolean	310
* 15.2.8 CallFunctionInteger64(paramcount as Integer,data as memoryblock) as boolean	311
* 15.2.9 CallMethod(param as string,data as memoryblock) as boolean	311
* 15.2.10 CallMethod(paramcount as Integer,data as memoryblock) as boolean	311
* 15.2.11 CopyLibrary(byref target as SoftDeclareMBS)	312
* 15.2.12 FreeLibrary as boolean	312
* 15.2.13 LoadConstant(constname as string) as boolean	312
* 15.2.14 LoadDLL(libname as string) as boolean	313
* 15.2.15 LoadDLLfromMemory(data as string) as boolean	313
* 15.2.16 LoadDylib(path as string) as boolean	314
* 15.2.17 LoadFramework(frameworkfilename as string) as boolean	314
* 15.2.18 LoadFrameworkFile(frameworkpath as folderitem) as boolean	315
* 15.2.19 LoadFunction(funcname as string) as boolean	315
* 15.2.20 LoadLibrary(libname as string) as boolean	316
* 15.2.21 ParametersSupported(param as string) as boolean	317
* 15.2.23 CallingMode as Integer	317
* 15.2.24 ConstantFound as boolean	317
* 15.2.25 ConstantName as string	317
* 15.2.26 ConstantPointer as Integer	318
* 15.2.27 FunctionFound as boolean	318
* 15.2.28 FunctionName as string	318
* 15.2.29 FunctionPointer as Integer	318
* 15.2.30 Lasterror as Integer	318
* 15.2.31 Liberror as string	319
* 15.2.32 Libfound as boolean	319
* 15.2.33 Libhandle as Integer	319
* 15.2.34 Libname as string	319
* 15.2.35 Result as Integer	319
* 15.2.36 ResultDouble as Double	320
* 15.2.37 ResultInt64 as MemoryBlock	320

	103
• 35 Sort	737
– 35.1.1 module SortMBS	737
* 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean)	738
* 35.1.4 AddArrayMBS(array1() as Color, array2() as Color)	739
* 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency)	739
* 35.1.6 AddArrayMBS(array1() as Date, array2() as Date)	740
* 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime)	740
* 35.1.8 AddArrayMBS(array1() as Double, array2() as Double)	741
* 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32)	741
* 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64)	742
* 35.1.11 AddArrayMBS(array1() as Object, array2() as Object)	742
* 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr)	743
* 35.1.13 AddArrayMBS(array1() as Single, array2() as Single)	743
* 35.1.14 AddArrayMBS(array1() as String, array2() as String)	744
* 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32)	744
* 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64)	745
* 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant)	745
* 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean)	746
* 35.1.19 AddMBS(extends array1() as Color, array2() as Color)	746
* 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency)	747
* 35.1.21 AddMBS(extends array1() as Date, array2() as Date)	747
* 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime)	748
* 35.1.23 AddMBS(extends array1() as Double, array2() as Double)	748
* 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32)	749
* 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64)	749
* 35.1.26 AddMBS(extends array1() as Object, array2() as Object)	750
* 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr)	751
* 35.1.28 AddMBS(extends array1() as Single, array2() as Single)	751
* 35.1.29 AddMBS(extends array1() as String, array2() as String)	752
* 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32)	752
* 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64)	753
* 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant)	753
* 35.1.33 CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)	754
* 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)	754
* 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)	755
* 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)	756
* 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)	757

- * 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- * 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- * 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- * 35.1.41 CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 761
- * 35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- * 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- * 35.1.44 CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 764
- * 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- * 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- * 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- * 35.1.48 CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 768
- * 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- * 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- * 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- * 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- * 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- * 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- * 35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- * 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- * 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- * 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- * 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778

- * 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- * 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780
- * 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781
- * 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- * 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- * 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- * 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- * 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- * 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- * 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- * 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- * 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- * 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789
- * 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- * 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- * 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- * 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- * 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- * 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- * 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- * 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- * 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798

* 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	800
* 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	800
* 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	801
* 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	802
* 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	803
* 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	804
* 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	805
* 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	806
* 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	807
* 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	808
* 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	809
* 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	810
* 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	811
* 35.1.95 ReverseMBS(extends theArray() as Boolean)	812
* 35.1.96 ReverseMBS(extends theArray() as Color)	813
* 35.1.97 ReverseMBS(extends theArray() as Currency)	813
* 35.1.98 ReverseMBS(extends theArray() as Date)	814
* 35.1.99 ReverseMBS(extends theArray() as DateTime)	814
* 35.1.100 ReverseMBS(extends theArray() as Double)	815
* 35.1.101 ReverseMBS(extends theArray() as Int32)	815
* 35.1.102 ReverseMBS(extends theArray() as Int64)	816
* 35.1.103 ReverseMBS(extends theArray() as Object)	817
* 35.1.104 ReverseMBS(extends theArray() as Ptr)	817
* 35.1.105 ReverseMBS(extends theArray() as Single)	818
* 35.1.106 ReverseMBS(extends theArray() as String)	818
* 35.1.107 ReverseMBS(extends theArray() as UInt32)	819
* 35.1.108 ReverseMBS(extends theArray() as UInt64)	819
* 35.1.109 ReverseMBS(extends theArray() as Variant)	820
* 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false)	820
* 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)	821

- * 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false) 822
- * 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- * 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- * 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 825
- * 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false) 825
- * 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- * 35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 828
- * 35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false) 828
- * 35.1.120 SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false) 829
- * 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830
- * 35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 831
- * 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false) 831
- * 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- * 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- * 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- * 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835
- * 35.1.128 SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 836
- * 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- * 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 837
- * 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false) 838
- * 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 839
- * 35.1.133 SortArrayMBS(theArray() as UInt64, descending as boolean = false) 840
- * 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 841
- * 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 841
- * 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842
- * 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- * 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- * 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- * 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845

* 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	846
* 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false)	846
* 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false)	847
* 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)	848
* 35.1.145 SortMBS(extends theArray() as Double, descending as boolean = false)	848
* 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	850
* 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)	851
* 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)	851
* 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)	852
* 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)	853
* 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false)	854
* 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)	854
* 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false)	855
* 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	856
* 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false)	857
* 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)	857
* 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false)	858
* 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)	859
* 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false)	860
* 35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)	860
* 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)	861
* 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64	862
* 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency	862
* 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double	864
* 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64	865
* 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64	866
* 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double	867

- * 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868
- * 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868
- * 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- * 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- * 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871
- * 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
- * 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
- * 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
- * 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
- * 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874
- * 35.1.179 SortVariantDelegateBooleanMBS(v1 as Boolean, v2 as Boolean) as Integer 875
- * 35.1.180 SortVariantDelegateColorMBS(v1 as Color, v2 as Color) as Integer 875
- * 35.1.181 SortVariantDelegateCurrencyMBS(v1 as Currency, v2 as Currency) as Integer 875
- * 35.1.182 SortVariantDelegateDateTimeMBS(v1 as DateTime, v2 as DateTime) as Integer 875
- * 35.1.183 SortVariantDelegateDoubleMBS(v1 as Double, v2 as Double) as Integer 876
- * 35.1.184 SortVariantDelegateInt32MBS(v1 as Int32, v2 as Int32) as Integer 876
- * 35.1.185 SortVariantDelegateInt64MBS(v1 as Int64, v2 as Int64) as Integer 876
- * 35.1.186 SortVariantDelegatePtrMBS(v1 as Ptr, v2 as Ptr) as Integer 877
- * 35.1.187 SortVariantDelegateSingleMBS(v1 as Single, v2 as Single) as Integer 877
- * 35.1.188 SortVariantDelegateStringMBS(v1 as String, v2 as String) as Integer 877
- * 35.1.189 SortVariantDelegateUInt32MBS(v1 as UInt32, v2 as UInt32) as Integer 878
- * 35.1.190 SortVariantDelegateUInt64MBS(v1 as UInt64, v2 as UInt64) as Integer 878
- * 35.1.191 SortVariantDelegateVariantMBS(v1 as Variant, v2 as Variant) as Integer 878

• 6 Audio	211
– ?? Globals	??
* 6.1.1 GetSoundMuteMBS as boolean	211
* 6.1.2 GetSoundVolumeLeftMBS as Double	211
* 6.1.3 GetSoundVolumeMBS as Double	211
* 6.1.4 GetSoundVolumeRightMBS as Double	212
* 6.1.5 SetSoundMuteMBS(mute as boolean)	212
* 6.1.6 SetSoundVolumeLeftMBS(percent as Double)	212
* 6.1.7 SetSoundVolumeMBS(percent as Double)	212
* 6.1.8 SetSoundVolumeRightMBS(percent as Double)	213

	111
• 36 Spamsum	879
– 36.1.1 class SpamSumMBS	879
* 36.1.3 Match(sum1 as string, sum2 as string) as Integer	881
* 36.1.4 Spamsum(text as string, flags as Integer, blocksize as Integer) as string	882
* 36.1.6 FlagsIgnoreHeaders as Integer	882
* 36.1.7 FlagsIgnoreWhitespace as Integer	882

• 37 Special Folders	883
– 37.1 Globals	883
* 37.1.1 ALMLocationsFolderMBS(domain as Integer) as folderitem	883
* 37.1.2 ALMModulesFolderMBS(domain as Integer) as folderitem	884
* 37.1.3 ALMPreferencesFolderMBS(domain as Integer) as folderitem	884
* 37.1.4 AppearanceFolderMBS(domain as Integer) as folderitem	885
* 37.1.5 AppleExtrasFolderMBS(domain as Integer) as folderitem	885
* 37.1.6 AppleMenuFolderMBS(domain as Integer) as folderitem	886
* 37.1.7 AppleShareAuthenticationFolderMBS(domain as Integer) as folderitem	887
* 37.1.8 AppleshareAutomountServerAliasesFolderMBS(domain as Integer) as folderitem	887
* 37.1.9 AppleShareSupportFolderMBS(domain as Integer) as folderitem	888
* 37.1.10 ApplicationsFolderMBS(domain as Integer) as folderitem	889
* 37.1.11 ApplicationSupportFolderMBS(domain as Integer) as folderitem	889
* 37.1.12 AssistantsFolderMBS(domain as Integer) as folderitem	890
* 37.1.13 AudioAlertSoundsFolderMBS(domain as Integer) as folderitem	891
* 37.1.14 AudioComponentsFolderMBS(domain as Integer) as folderitem	892
* 37.1.15 AudioDigidesignFolderMBS(domain as Integer) as folderitem	892
* 37.1.16 AudioPlugInsFolderMBS(domain as Integer) as folderitem	893
* 37.1.17 AudioPresetsFolderMBS(domain as Integer) as folderitem	893
* 37.1.18 AudioSoundBanksFolderMBS(domain as Integer) as folderitem	894
* 37.1.19 AudioSoundsFolderMBS(domain as Integer) as folderitem	895
* 37.1.20 AudioSupportFolderMBS(domain as Integer) as folderitem	895
* 37.1.21 AudioVSTFolderMBS(domain as Integer) as folderitem	896
* 37.1.22 AutomatorWorkflowsFolderMBS(domain as Integer) as folderitem	897
* 37.1.23 AutosaveInformationFolderMBS(domain as Integer) as folderitem	897
* 37.1.24 BootTimeStartupItemsFolderMBS(domain as Integer) as folderitem	898
* 37.1.25 CachedDataFolderMBS(domain as Integer) as folderitem	899
* 37.1.26 CarbonLibraryFolderMBS(domain as Integer) as folderitem	899
* 37.1.27 ChewableItemsFolderMBS(domain as Integer) as folderitem	900
* 37.1.28 classicDesktopFolderMBS(domain as Integer) as folderitem	901
* 37.1.29 ClassicPreferencesFolderMBS(domain as Integer) as folderitem	901
* 37.1.30 ColorPickersFolderMBS(domain as Integer) as folderitem	902
* 37.1.31 ColorSyncCMMFolderMBS(domain as Integer) as folderitem	903
* 37.1.32 ColorSyncFolderMBS(domain as Integer) as folderitem	903
* 37.1.33 ColorSyncProfilesFolderMBS(domain as Integer) as folderitem	904
* 37.1.34 ColorSyncScriptingFolderMBS(domain as Integer) as folderitem	904
* 37.1.35 ComponentsFolderMBS(domain as Integer) as folderitem	905
* 37.1.36 CompositionsFolderMBS(domain as Integer) as folderitem	906
* 37.1.37 ContextualMenuItemsFolderMBS(domain as Integer) as folderitem	906
* 37.1.38 ControlPanelDisabledFolderMBS(domain as Integer) as folderitem	907
* 37.1.39 ControlPanelFolderMBS(domain as Integer) as folderitem	908

* 37.1.40 ControlStripModulesFolderMBS(domain as Integer) as folderitem	908
* 37.1.41 CoreServicesFolderMBS(domain as Integer) as folderitem	909
* 37.1.42 CreateALMLocationsFolderMBS(domain as Integer) as folderitem	909
* 37.1.43 CreateALMModulesFolderMBS(domain as Integer) as folderitem	910
* 37.1.44 CreateALMPreferencesFolderMBS(domain as Integer) as folderitem	911
* 37.1.45 CreateAppearanceFolderMBS(domain as Integer) as folderitem	911
* 37.1.46 CreateAppleExtrasFolderMBS(domain as Integer) as folderitem	912
* 37.1.47 CreateAppleMenuFolderMBS(domain as Integer) as folderitem	913
* 37.1.48 CreateAppleShareAuthenticationFolderMBS(domain as Integer) as folderitem	913
* 37.1.49 CreateAppleshareAutomountServerAliasesFolderMBS(domain as Integer) as folderitem	914
* 37.1.50 CreateAppleShareSupportFolderMBS(domain as Integer) as folderitem	915
* 37.1.51 CreateApplicationsFolderMBS(domain as Integer) as folderitem	915
* 37.1.52 CreateApplicationSupportFolderMBS(domain as Integer) as folderitem	916
* 37.1.53 CreateAssistantsFolderMBS(domain as Integer) as folderitem	917
* 37.1.54 CreateAudioAlertSoundsFolderMBS(domain as Integer) as folderitem	918
* 37.1.55 CreateAudioComponentsFolderMBS(domain as Integer) as folderitem	918
* 37.1.56 CreateAudioDigidesignFolderMBS(domain as Integer) as folderitem	919
* 37.1.57 CreateAudioPlugInsFolderMBS(domain as Integer) as folderitem	920
* 37.1.58 CreateAudioPresetsFolderMBS(domain as Integer) as folderitem	920
* 37.1.59 CreateAudioSoundBanksFolderMBS(domain as Integer) as folderitem	921
* 37.1.60 CreateAudioSoundsFolderMBS(domain as Integer) as folderitem	922
* 37.1.61 CreateAudioSupportFolderMBS(domain as Integer) as folderitem	922
* 37.1.62 CreateAudioVSTFolderMBS(domain as Integer) as folderitem	923
* 37.1.63 CreateAutomatorWorkflowsFolderMBS(domain as Integer) as folderitem	924
* 37.1.64 CreateAutosaveInformationFolderMBS(domain as Integer) as folderitem	924
* 37.1.65 CreateBootTimeStartupItemsFolderMBS(domain as Integer) as folderitem	925
* 37.1.66 CreateCachedDataFolderMBS(domain as Integer) as folderitem	926
* 37.1.67 CreateCarbonLibraryFolderMBS(domain as Integer) as folderitem	926
* 37.1.68 CreateChewableItemsFolderMBS(domain as Integer) as folderitem	927
* 37.1.69 CreateClassicDesktopFolderMBS(domain as Integer) as folderitem	928
* 37.1.70 CreateClassicPreferencesFolderMBS(domain as Integer) as folderitem	928
* 37.1.71 CreateColorPickersFolderMBS(domain as Integer) as folderitem	929
* 37.1.72 CreateColorSyncCMMFolderMBS(domain as Integer) as folderitem	930
* 37.1.73 CreateColorSyncFolderMBS(domain as Integer) as folderitem	930
* 37.1.74 CreateColorSyncProfilesFolderMBS(domain as Integer) as folderitem	931
* 37.1.75 CreateColorSyncScriptingFolderMBS(domain as Integer) as folderitem	932
* 37.1.76 CreateComponentsFolderMBS(domain as Integer) as folderitem	932
* 37.1.77 CreateCompositionsFolderMBS(domain as Integer) as folderitem	933
* 37.1.78 CreateContextualMenuItemsFolderMBS(domain as Integer) as folderitem	934
* 37.1.79 CreateControlPanelDisabledFolderMBS(domain as Integer) as folderitem	934
* 37.1.80 CreateControlPanelFolderMBS(domain as Integer) as folderitem	935

* 37.1.81 CreateControlStripModulesFolderMBS(domain as Integer) as folderitem	936
* 37.1.82 CreateCoreServicesFolderMBS(domain as Integer) as folderitem	936
* 37.1.83 CreateCurrentUserFolderMBS(domain as Integer) as folderitem	937
* 37.1.84 CreateCurrentUserRemoteFolderLocationFolderMBS(domain as Integer) as folderitem	938
* 37.1.85 CreateCurrentUserRemoteFolderMBS(domain as Integer) as folderitem	938
* 37.1.86 CreateDesktopFolderMBS(domain as Integer) as folderitem	939
* 37.1.87 CreateDesktopPicturesFolderMBS(domain as Integer) as folderitem	940
* 37.1.88 CreateDeveloperApplicationsFolderMBS(domain as Integer) as folderitem	940
* 37.1.89 CreateDeveloperDocsFolderMBS(domain as Integer) as folderitem	941
* 37.1.90 CreateDeveloperFolderMBS(domain as Integer) as folderitem	942
* 37.1.91 CreateDeveloperHelpFolderMBS(domain as Integer) as folderitem	942
* 37.1.92 CreateDictionariesFolderMBS(domain as Integer) as folderitem	943
* 37.1.93 CreateDirectoryServicesFolderMBS(domain as Integer) as folderitem	944
* 37.1.94 CreateDirectoryServicesPlugInsFolderMBS(domain as Integer) as folderitem	944
* 37.1.95 CreateDisplayExtensionsFolderMBS(domain as Integer) as folderitem	945
* 37.1.96 CreateDocumentationFolderMBS(domain as Integer) as folderitem	946
* 37.1.97 CreateDocumentsFolderMBS(domain as Integer) as folderitem	946
* 37.1.98 CreateDomainLibraryFolderMBS(domain as Integer) as folderitem	947
* 37.1.99 CreateDomainTopLevelFolderMBS(domain as Integer) as folderitem	948
* 37.1.100 CreateDownloadsFolderMBS(domain as Integer) as folderitem	948
* 37.1.101 CreateEditorsFolderMBS(domain as Integer) as folderitem	949
* 37.1.102 CreateExtensionDisabledFolderMBS(domain as Integer) as folderitem	950
* 37.1.103 CreateExtensionFolderMBS(domain as Integer) as folderitem	950
* 37.1.104 CreateFavoritesFolderMBS(domain as Integer) as folderitem	951
* 37.1.105 CreateFileSystemSupportFolderMBS(domain as Integer) as folderitem	952
* 37.1.106 CreateFindByContentFolderMBS(domain as Integer) as folderitem	952
* 37.1.107 CreateFindByContentIndexesFolderMBS(domain as Integer) as folderitem	953
* 37.1.108 CreateFindByContentPluginsFolderMBS(domain as Integer) as folderitem	954
* 37.1.109 CreateFindSupportFolderMBS(domain as Integer) as folderitem	954
* 37.1.110 CreateFolderActionsFolderMBS(domain as Integer) as folderitem	955
* 37.1.111 CreateFontCollectionsFolderMBS(domain as Integer) as folderitem	956
* 37.1.112 CreateFontsFolderMBS(domain as Integer) as folderitem	956
* 37.1.113 CreateFrameworksFolderMBS(domain as Integer) as folderitem	957
* 37.1.114 CreateGenEditorsFolderMBS(domain as Integer) as folderitem	958
* 37.1.115 CreateHelpFolderMBS(domain as Integer) as folderitem	958
* 37.1.116 CreateiMovieFolderMBS(domain as Integer) as folderitem	959
* 37.1.117 CreateiMoviePlugInsFolderMBS(domain as Integer) as folderitem	960
* 37.1.118 CreateiMovieSoundEffectsFolderMBS(domain as Integer) as folderitem	960
* 37.1.119 CreateIndexFilesFolderMBS(domain as Integer) as folderitem	961
* 37.1.120 CreateInputManagersFolderMBS(domain as Integer) as folderitem	962
* 37.1.121 CreateInputMethodsFolderMBS(domain as Integer) as folderitem	962

* 37.1.122 CreateInstallerLogsFolderMBS(domain as Integer) as folderitem	963
* 37.1.123 CreateInstallerReceiptsFolderMBS(domain as Integer) as folderitem	964
* 37.1.124 CreateInternetFolderMBS(domain as Integer) as folderitem	964
* 37.1.125 CreateInternetPlugInFolderMBS(domain as Integer) as folderitem	965
* 37.1.126 CreateInternetSearchSitesFolderMBS(domain as Integer) as folderitem	966
* 37.1.127 CreateInternetSitesFolderMBS(domain as Integer) as folderitem	966
* 37.1.128 CreateISSDownloadsFolderMBS(domain as Integer) as folderitem	967
* 37.1.129 CreateKernelExtensionsFolderMBS(domain as Integer) as folderitem	968
* 37.1.130 CreateKeyboardLayoutsFolderMBS(domain as Integer) as folderitem	968
* 37.1.131 CreateKeychainFolderMBS(domain as Integer) as folderitem	969
* 37.1.132 CreateLauncherItemsFolderMBS(domain as Integer) as folderitem	970
* 37.1.133 CreateLibraryAssistantsFolderMBS(domain as Integer) as folderitem	970
* 37.1.134 CreateLocalesFolderMBS(domain as Integer) as folderitem	971
* 37.1.135 CreateLogsFolderMBS(domain as Integer) as folderitem	972
* 37.1.136 CreateMacOSReadMesFolderMBS(domain as Integer) as folderitem	972
* 37.1.137 CreateMagicTemporaryItemsFolderMBS(domain as Integer) as folderitem	973
* 37.1.138 CreateManagedItemsFolderMBS(domain as Integer) as folderitem	974
* 37.1.139 CreateMIDIDriversFolderMBS(domain as Integer) as folderitem	974
* 37.1.140 CreateModemScriptsFolderMBS(domain as Integer) as folderitem	975
* 37.1.141 CreateMovieDocumentsFolderMBS(domain as Integer) as folderitem	976
* 37.1.142 CreateMultiprocessingFolderMBS(domain as Integer) as folderitem	976
* 37.1.143 CreateMusicDocumentsFolderMBS(domain as Integer) as folderitem	977
* 37.1.144 CreateOpenDocEditorsFolderMBS(domain as Integer) as folderitem	978
* 37.1.145 CreateOpenDocFolderMBS(domain as Integer) as folderitem	978
* 37.1.146 CreateOpenDocLibrariesFolderMBS(domain as Integer) as folderitem	979
* 37.1.147 CreateOpenDocShellPlugInsFolderMBS(domain as Integer) as folderitem	980
* 37.1.148 CreatePictureDocumentsFolderMBS(domain as Integer) as folderitem	980
* 37.1.149 CreatePreferencePanelsFolderMBS(domain as Integer) as folderitem	981
* 37.1.150 CreatePreferencesFolderMBS(domain as Integer) as folderitem	982
* 37.1.151 CreatePrinterDescriptionFolderMBS(domain as Integer) as folderitem	982
* 37.1.152 CreatePrinterDriverFolderMBS(domain as Integer) as folderitem	983
* 37.1.153 CreatePrintersFolderMBS(domain as Integer) as folderitem	984
* 37.1.154 CreatePrintingPlugInsFolderMBS(domain as Integer) as folderitem	984
* 37.1.155 CreatePrintMonitorDocsFolderMBS(domain as Integer) as folderitem	985
* 37.1.156 CreatePrivateFrameworksFolderMBS(domain as Integer) as folderitem	986
* 37.1.157 CreatePublicFolderMBS(domain as Integer) as folderitem	986
* 37.1.158 CreateQuickLookFolderMBS(domain as Integer) as folderitem	987
* 37.1.159 CreateQuickTimeComponentsFolderMBS(domain as Integer) as folderitem	988
* 37.1.160 CreateQuickTimeExtensionsFolderMBS(domain as Integer) as folderitem	988
* 37.1.161 CreateRecentApplicationsFolderMBS(domain as Integer) as folderitem	989
* 37.1.162 CreateRecentDocumentsFolderMBS(domain as Integer) as folderitem	990
* 37.1.163 CreateRecentServersFolderMBS(domain as Integer) as folderitem	990

* 37.1.164 CreateScriptingAdditionsFolderMBS(domain as Integer) as folderitem	991
* 37.1.165 CreateScriptsFolderMBS(domain as Integer) as folderitem	992
* 37.1.166 CreateSharedLibrariesFolderMBS(domain as Integer) as folderitem	993
* 37.1.167 CreateSharedUserDataFolderMBS(domain as Integer) as folderitem	993
* 37.1.168 CreateShutdownFolderMBS(domain as Integer) as folderitem	994
* 37.1.169 CreateShutdownItemsDisabledFolderMBS(domain as Integer) as folderitem	995
* 37.1.170 CreateSoundSetsFolderMBS(domain as Integer) as folderitem	995
* 37.1.171 CreateSpeakableItemsFolderMBS(domain as Integer) as folderitem	996
* 37.1.172 CreateSpeechFolderMBS(domain as Integer) as folderitem	997
* 37.1.173 CreateSpotlightImportersFolderMBS(domain as Integer) as folderitem	997
* 37.1.174 CreateSpotlightMetadataCacheFolderMBS(domain as Integer) as folderitem	998
* 37.1.175 CreateSpotlightSavedSearchesFolderMBS(domain as Integer) as folderitem	999
* 37.1.176 CreateStartupFolderMBS(domain as Integer) as folderitem	999
* 37.1.177 CreateStartupItemsDisabledFolderMBS(domain as Integer) as folderitem	1000
* 37.1.178 CreateStationeryFolderMBS(domain as Integer) as folderitem	1001
* 37.1.179 CreateSystemControlPanelFolderMBS(domain as Integer) as folderitem	1001
* 37.1.180 CreateSystemDesktopFolderMBS(domain as Integer) as folderitem	1002
* 37.1.181 CreateSystemExtensionDisabledFolderMBS(domain as Integer) as folderitem	1003
* 37.1.182 CreateSystemFolderMBS(domain as Integer) as folderitem	1003
* 37.1.183 CreateSystemPreferencesFolderMBS(domain as Integer) as folderitem	1004
* 37.1.184 CreateSystemSoundsFolderMBS(domain as Integer) as folderitem	1005
* 37.1.185 CreateSystemTrashFolderMBS(domain as Integer) as folderitem	1005
* 37.1.186 CreateTemporaryFolderMBS(domain as Integer) as folderitem	1006
* 37.1.187 CreateTemporaryItemsInCacheDataFolderMBS(domain as Integer) as folderitem	1007
* 37.1.188 CreateTemporaryItemsInUserDomainFolderMBS(domain as Integer) as folderitem	1007
* 37.1.189 CreateTextEncodingsFolderMBS(domain as Integer) as folderitem	1008
* 37.1.190 CreateThemesFolderMBS(domain as Integer) as folderitem	1009
* 37.1.191 CreateTrashFolderMBS(domain as Integer) as folderitem	1009
* 37.1.192 CreateUsersFolderMBS(domain as Integer) as folderitem	1010
* 37.1.193 CreateUserSpecificTmpFolderMBS(domain as Integer) as folderitem	1011
* 37.1.194 CreateUtilitiesFolderMBS(domain as Integer) as folderitem	1011
* 37.1.195 CreateVoicesFolderMBS(domain as Integer) as folderitem	1012
* 37.1.196 CreateVolumeRootFolderMBS(domain as Integer) as folderitem	1013
* 37.1.197 CreateVolumeSettingsFolderMBS(domain as Integer) as folderitem	1013
* 37.1.198 CreateWhereToEmptyTrashFolderMBS(domain as Integer) as folderitem	1014
* 37.1.199 CurrentUserFolderMBS(domain as Integer) as folderitem	1015
* 37.1.200 CurrentUserRemoteFolderLocationFolderMBS(domain as Integer) as folderitem	1015
* 37.1.201 CurrentUserRemoteFolderMBS(domain as Integer) as folderitem	1016
* 37.1.202 DesktopFolderMBS(domain as Integer) as folderitem	1016
* 37.1.203 DesktopPicturesFolderMBS(domain as Integer) as folderitem	1017
* 37.1.204 DeveloperApplicationsFolderMBS(domain as Integer) as folderitem	1018

* 37.1.205 DeveloperDocsFolderMBS(domain as Integer) as folderitem	1018
* 37.1.206 DeveloperFolderMBS(domain as Integer) as folderitem	1019
* 37.1.207 DeveloperHelpFolderMBS(domain as Integer) as folderitem	1019
* 37.1.208 DictionariesFolderMBS(domain as Integer) as folderitem	1020
* 37.1.209 DirectoryServicesFolderMBS(domain as Integer) as folderitem	1020
* 37.1.210 DirectoryServicesPlugInsFolderMBS(domain as Integer) as folderitem	1021
* 37.1.211 DisplayExtensionsFolderMBS(domain as Integer) as folderitem	1022
* 37.1.212 DocumentationFolderMBS(domain as Integer) as folderitem	1022
* 37.1.213 DocumentsFolderMBS(domain as Integer) as folderitem	1023
* 37.1.214 DomainLibraryFolderMBS(domain as Integer) as folderitem	1023
* 37.1.215 DomainTopLevelFolderMBS(domain as Integer) as folderitem	1024
* 37.1.216 DownloadsFolderMBS(domain as Integer) as folderitem	1025
* 37.1.217 EditorsFolderMBS(domain as Integer) as folderitem	1026
* 37.1.218 ExtensionDisabledFolderMBS(domain as Integer) as folderitem	1026
* 37.1.219 ExtensionFolderMBS(domain as Integer) as folderitem	1027
* 37.1.220 FavoritesFolderMBS(domain as Integer) as folderitem	1027
* 37.1.221 FileSystemSupportFolderMBS(domain as Integer) as folderitem	1028
* 37.1.222 FindByContentFolderMBS(domain as Integer) as folderitem	1028
* 37.1.223 FindByContentIndexesFolderMBS(domain as Integer) as folderitem	1029
* 37.1.224 FindByContentPluginsFolderMBS(domain as Integer) as folderitem	1030
* 37.1.225 FindSupportFolderMBS(domain as Integer) as folderitem	1031
* 37.1.226 FolderActionsFolderMBS(domain as Integer) as folderitem	1031
* 37.1.227 FontCollectionsFolderMBS(domain as Integer) as folderitem	1032
* 37.1.228 FontsFolderMBS(domain as Integer) as folderitem	1032
* 37.1.229 FrameworksFolderMBS(domain as Integer) as folderitem	1033
* 37.1.230 GenEditorsFolderMBS(domain as Integer) as folderitem	1033
* 37.1.231 HelpFolderMBS(domain as Integer) as folderitem	1034
* 37.1.232 iMovieFolderMBS(domain as Integer) as folderitem	1035
* 37.1.233 iMoviePlugInsFolderMBS(domain as Integer) as folderitem	1036
* 37.1.234 iMovieSoundEffectsFolderMBS(domain as Integer) as folderitem	1036
* 37.1.235 IndexFilesFolderMBS(domain as Integer) as folderitem	1037
* 37.1.236 InputManagersFolderMBS(domain as Integer) as folderitem	1038
* 37.1.237 InputMethodsFolderMBS(domain as Integer) as folderitem	1038
* 37.1.238 InstallerLogsFolderMBS(domain as Integer) as folderitem	1039
* 37.1.239 InstallerReceiptsFolderMBS(domain as Integer) as folderitem	1040
* 37.1.240 InternetFolderMBS(domain as Integer) as folderitem	1040
* 37.1.241 InternetPlugInFolderMBS(domain as Integer) as folderitem	1041
* 37.1.242 InternetSearchSitesFolderMBS(domain as Integer) as folderitem	1041
* 37.1.243 InternetSitesFolderMBS(domain as Integer) as folderitem	1042
* 37.1.244 ISSDownloadsFolderMBS(domain as Integer) as folderitem	1042
* 37.1.245 KernelExtensionsFolderMBS(domain as Integer) as folderitem	1043
* 37.1.246 KeyboardLayoutsFolderMBS(domain as Integer) as folderitem	1044

* 37.1.247 KeychainFolderMBS(domain as Integer) as folderitem	1045
* 37.1.248 LauncherItemsFolderMBS(domain as Integer) as folderitem	1045
* 37.1.249 LibraryAssistantsFolderMBS(domain as Integer) as folderitem	1046
* 37.1.250 LocalesFolderMBS(domain as Integer) as folderitem	1046
* 37.1.251 LogsFolderMBS(domain as Integer) as folderitem	1047
* 37.1.252 MacOSReadMesFolderMBS(domain as Integer) as folderitem	1047
* 37.1.253 MagicTemporaryItemsFolderMBS(domain as Integer) as folderitem	1048
* 37.1.254 ManagedItemsFolderMBS(domain as Integer) as folderitem	1049
* 37.1.255 MIDIDriversFolderMBS(domain as Integer) as folderitem	1050
* 37.1.256 ModemScriptsFolderMBS(domain as Integer) as folderitem	1050
* 37.1.257 MovieDocumentsFolderMBS(domain as Integer) as folderitem	1051
* 37.1.258 MultiprocessingFolderMBS(domain as Integer) as folderitem	1051
* 37.1.259 MusicDocumentsFolderMBS(domain as Integer) as folderitem	1052
* 37.1.260 OpenDocEditorsFolderMBS(domain as Integer) as folderitem	1052
* 37.1.261 OpenDocFolderMBS(domain as Integer) as folderitem	1053
* 37.1.262 OpenDocLibrariesFolderMBS(domain as Integer) as folderitem	1054
* 37.1.263 OpenDocShellPlugInsFolderMBS(domain as Integer) as folderitem	1055
* 37.1.264 PictureDocumentsFolderMBS(domain as Integer) as folderitem	1055
* 37.1.265 PreferencePanelsFolderMBS(domain as Integer) as folderitem	1056
* 37.1.266 PreferencesFolderMBS(domain as Integer) as folderitem	1056
* 37.1.267 PrinterDescriptionFolderMBS(domain as Integer) as folderitem	1057
* 37.1.268 PrinterDriverFolderMBS(domain as Integer) as folderitem	1058
* 37.1.269 PrintersFolderMBS(domain as Integer) as folderitem	1058
* 37.1.270 PrintingPlugInsFolderMBS(domain as Integer) as folderitem	1059
* 37.1.271 PrintMonitorDocsFolderMBS(domain as Integer) as folderitem	1059
* 37.1.272 PrivateFrameworksFolderMBS(domain as Integer) as folderitem	1060
* 37.1.273 PublicFolderMBS(domain as Integer) as folderitem	1061
* 37.1.274 QuickLookFolderMBS(domain as Integer) as folderitem	1061
* 37.1.275 QuickTimeComponentsFolderMBS(domain as Integer) as folderitem	1062
* 37.1.276 QuickTimeExtensionsFolderMBS(domain as Integer) as folderitem	1063
* 37.1.277 RecentApplicationsFolderMBS(domain as Integer) as folderitem	1063
* 37.1.278 RecentDocumentsFolderMBS(domain as Integer) as folderitem	1064
* 37.1.279 RecentServersFolderMBS(domain as Integer) as folderitem	1065
* 37.1.280 ScriptingAdditionsFolderMBS(domain as Integer) as folderitem	1065
* 37.1.281 ScriptsFolderMBS(domain as Integer) as folderitem	1066
* 37.1.282 SharedLibrariesFolderMBS(domain as Integer) as folderitem	1066
* 37.1.283 SharedUserDataFolderMBS(domain as Integer) as folderitem	1067
* 37.1.284 ShutdownFolderMBS(domain as Integer) as folderitem	1067
* 37.1.285 ShutdownItemsDisabledFolderMBS(domain as Integer) as folderitem	1068
* 37.1.286 SoundSetsFolderMBS(domain as Integer) as folderitem	1069
* 37.1.287 SpeakableItemsFolderMBS(domain as Integer) as folderitem	1069
* 37.1.288 SpeechFolderMBS(domain as Integer) as folderitem	1070

* 37.1.289 SpotlightImportersFolderMBS(domain as Integer) as folderitem	1070
* 37.1.290 SpotlightMetadataCacheFolderMBS(domain as Integer) as folderitem	1071
* 37.1.291 SpotlightSavedSearchesFolderMBS(domain as Integer) as folderitem	1072
* 37.1.292 StartupFolderMBS(domain as Integer) as folderitem	1073
* 37.1.293 StartupItemsDisabledFolderMBS(domain as Integer) as folderitem	1073
* 37.1.294 StationeryFolderMBS(domain as Integer) as folderitem	1074
* 37.1.295 SystemControlPanelFolderMBS(domain as Integer) as folderitem	1075
* 37.1.296 SystemDesktopFolderMBS(domain as Integer) as folderitem	1075
* 37.1.297 SystemExtensionDisabledFolderMBS(domain as Integer) as folderitem	1076
* 37.1.298 SystemFolderMBS(domain as Integer) as folderitem	1076
* 37.1.299 SystemPreferencesFolderMBS(domain as Integer) as folderitem	1077
* 37.1.300 SystemSoundsFolderMBS(domain as Integer) as folderitem	1077
* 37.1.301 SystemTrashFolderMBS(domain as Integer) as folderitem	1078
* 37.1.302 TemporaryFolderMBS(domain as Integer) as folderitem	1079
* 37.1.303 TemporaryItemsInCacheDataFolderMBS(domain as Integer) as folderitem	1079
* 37.1.304 TemporaryItemsInUserDomainFolderMBS(domain as Integer) as folderitem	1080
* 37.1.305 TextEncodingsFolderMBS(domain as Integer) as folderitem	1081
* 37.1.306 ThemesFolderMBS(domain as Integer) as folderitem	1081
* 37.1.307 TrashFolderMBS(domain as Integer) as folderitem	1082
* 37.1.308 UsersFolderMBS(domain as Integer) as folderitem	1082
* 37.1.309 UserSpecificTmpFolderMBS(domain as Integer) as folderitem	1083
* 37.1.310 UtilitiesFolderMBS(domain as Integer) as folderitem	1084
* 37.1.311 VoicesFolderMBS(domain as Integer) as folderitem	1084
* 37.1.312 VolumeRootFolderMBS(domain as Integer) as folderitem	1085
* 37.1.313 VolumeSettingsFolderMBS(domain as Integer) as folderitem	1085
* 37.1.314 WhereToEmptyTrashFolderMBS(domain as Integer) as folderitem	1086
* 37.1.315 WindowsBurnAreaFolderMBS as folderitem	1087
* 37.1.316 WindowsFolderMBS as folderitem	1087
* 37.1.317 WindowsSystemFolderMBS as folderitem	1087

• 26 Math	535
– 26.2.1 class SplineMBS	559
* 26.2.3 a(index as Integer) as Double	559
* 26.2.4 b(index as Integer) as Double	559
* 26.2.5 c(index as Integer) as Double	559
* 26.2.6 calc(x as Double) as Double	559
* 26.2.7 Constructor(X() as Double, Y() as Double)	560
* 26.2.8 d(index as Integer) as Double	560
* 26.2.9 x(index as Integer) as Double	560
* 26.2.10 y(index as Integer) as Double	560
* 26.2.12 count as Integer	560

	121
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

	123
* 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
* 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
* 38.1.60 StringORMBS(a as string,b as string) as string	1124
* 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.63 StrMBS(d as Double) as string	1126
* 38.1.64 UnicodeStringMBS(s as string) as string	1126

• 21 Files	379
– 21.7.1 class StdinMBS	449
* 21.7.3 AttachConsole(ProcessID as Integer = -1) as Integer	450
* 21.7.4 Flush	450
* 21.7.5 FreeConsole as Integer	450
* 21.7.6 GetCharacter as Integer	451
* 21.7.7 Read(count as Integer) as string	451
* 21.7.8 ReadDouble(byref value as Double) as Integer	451
* 21.7.9 ReadInteger(byref value as Integer) as Integer	451
* 21.7.10 ReadString(byref value as string) as Integer	451
* 21.7.12 Echo as Boolean	452
* 21.7.13 IsReady as boolean	452
– 21.8.1 class StdoutMBS	454
* 21.8.3 AttachConsole(ProcessID as Integer = -1) as Integer	455
* 21.8.4 Flush	455
* 21.8.5 FreeConsole as Integer	455
* 21.8.6 Write(data as string)	456

	125
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

	127
* 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
* 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
* 38.1.60 StringORMBS(a as string,b as string) as string	1124
* 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.63 StrMBS(d as Double) as string	1126
* 38.1.64 UnicodeStringMBS(s as string) as string	1126

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock, len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock, len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	129
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

- * 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean 1124
- * 38.1.59 StringIsXMLreadyMBS(s as string) as boolean 1124
- * 38.1.60 StringORMBS(a as string,b as string) as string 1124
- * 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.63 StrMBS(d as Double) as string 1126
- * 38.1.64 UnicodeStringMBS(s as string) as string 1126

• 44 XojoRuntime	1237
– ?? Globals	??
* 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()	1237
* 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()	1237
* 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()	1238
* 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()	1239
* 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean	1239
* 44.1.6 ArrayStringMBS(paramArray values as String) as String()	1240
* 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()	1240
* 44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer	1241
* 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
* 44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer	1242
* 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
* 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
* 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
* 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
* 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
* 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
* 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
* 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
* 44.1.18 GetTextMemoryAddressMBS(s as text) as integer	1246
* 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()	1247
* 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer	1248
* 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant	1248
* 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()	1249
* 44.1.23 GetVariantTypeMBS(va as variant) as Integer	1250
* 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
* 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)	1253
* 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)	1251

	133
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

- * 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean 1124
- * 38.1.59 StringIsXMLreadyMBS(s as string) as boolean 1124
- * 38.1.60 StringORMBS(a as string,b as string) as string 1124
- * 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.63 StrMBS(d as Double) as string 1126
- * 38.1.64 UnicodeStringMBS(s as string) as string 1126

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock, len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock, len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	137
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

- * 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean 1124
- * 38.1.59 StringIsXMLreadyMBS(s as string) as boolean 1124
- * 38.1.60 StringORMBS(a as string,b as string) as string 1124
- * 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.63 StrMBS(d as Double) as string 1126
- * 38.1.64 UnicodeStringMBS(s as string) as string 1126

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock, len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock, len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	141
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

	143
* 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
* 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
* 38.1.60 StringORMBS(a as string,b as string) as string	1124
* 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.63 StrMBS(d as Double) as string	1126
* 38.1.64 UnicodeStringMBS(s as string) as string	1126

• 44 XojoRuntime	1237
– ?? Globals	??
* 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()	1237
* 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()	1237
* 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()	1238
* 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()	1239
* 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean	1239
* 44.1.6 ArrayStringMBS(paramArray values as String) as String()	1240
* 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()	1240
* 44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer	1241
* 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
* 44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer	1242
* 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
* 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
* 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
* 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
* 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
* 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
* 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
* 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
* 44.1.18 GetTextMemoryAddressMBS(s as text) as integer	1246
* 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()	1247
* 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer	1248
* 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant	1248
* 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()	1249
* 44.1.23 GetVariantTypeMBS(va as variant) as Integer	1250
* 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
* 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)	1253
* 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)	1251

	145
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

	147
* 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
* 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
* 38.1.60 StringORMBS(a as string,b as string) as string	1124
* 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.63 StrMBS(d as Double) as string	1126
* 38.1.64 UnicodeStringMBS(s as string) as string	1126

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock, len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock, len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	149
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

	151
* 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
* 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
* 38.1.60 StringORMBS(a as string,b as string) as string	1124
* 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
* 38.1.63 StrMBS(d as Double) as string	1126
* 38.1.64 UnicodeStringMBS(s as string) as string	1126

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock,len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock,len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	153
• 38 String	1089
– ?? Globals	??
* 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
* 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string	1098
* 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099
* 38.1.14 ClearStringContentMBS(s as String) as Boolean	1100
* 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string	1100
* 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string	1101
* 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string	1101
* 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string	1102
* 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string	1102
* 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean	1102
* 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string	1126
* 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string	1127
* 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer	1104
* 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string	1104
* 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
* 38.1.68 DecodingFromHexMBS(s as string) as string	1128
* 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
* 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
* 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
* 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
* 38.1.26 DecodingFromURLMBS(s as string) as string	1106
* 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
* 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
* 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
* 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
* 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
* 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
* 38.1.71 EncodingToHexMBS(s as string) as string	1129
* 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
* 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
* 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
* 38.1.33 EncodingToURLMBS(s as string) as string	1111
* 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111
* 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string	1112
* 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112

- * 38.1.37 GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113
- * 38.1.38 GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113
- * 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string 1114
- * 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- * 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- * 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- * 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- * 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- * 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- * 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- * 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- * 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- * 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- * 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- * 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- * 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double 1117
- * 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double 1118
- * 38.1.47 NativeStringMBS(s as string) as string 1119
- * 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- * 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- * 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- * 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- * 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- * 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- * 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- * 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string() 1096
- * 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string 1122
- * 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer 1122
- * 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer 1123
- * 38.1.57 StringANDMBS(a as string,b as string) as string 1123

- * 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean 1124
- * 38.1.59 StringIsXMLreadyMBS(s as string) as boolean 1124
- * 38.1.60 StringORMBS(a as string,b as string) as string 1124
- * 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string 1125
- * 38.1.63 StrMBS(d as Double) as string 1126
- * 38.1.64 UnicodeStringMBS(s as string) as string 1126

- **26 Math** 535
 - 26.3.1 module SunTimesMBS 561
 - * 26.3.3 CalcJulianDate(day as Integer, month as Integer, year as Integer) as Double 562
 - * 26.3.4 CalcSunriseUTC(JD as Double, latitude as Double, longitude as Double) as Double 562
 - * 26.3.5 CalcSunsetUTC(JD as Double, latitude as Double, longitude as Double) as Double 562

	157
• 39 System	1143
– ?? Globals	??
* 39.1.6 AbortMBS	1146
* 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()	1147
* 39.1.1 CrashNiceMBS	1143
* 39.1.2 CrashUglyMBS	1143
* 39.1.3 DelayMBS(time as Double)	1143
* 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
* 39.1.7 ExitMBS(code as Integer)	1146
* 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
* 39.1.9 MillisecondsMBS as Double	1147
* 39.1.5 SleepMBS(time as Double)	1145
– 39.3.1 module SystemInformationMBS	1155
* 39.3.3 AvailableRAM as Double	1155
* 39.3.4 BusSpeed as Double	1156
* 39.3.5 Computername as string	1156
* 39.3.6 CPUBrandString as string	1157
* 39.3.7 CPUSpeed as Double	1157
* 39.3.8 DomainName as string	1157
* 39.3.9 HardDiscSerial as string	1158
* 39.3.10 HostName as string	1158
* 39.3.11 Is64bitWindows as boolean	1158
* 39.3.12 isARM as Boolean	1159
* 39.3.13 isBigSur(orHigher as boolean = true) as boolean	1159
* 39.3.14 isCatalina(orHigher as boolean = true) as boolean	1160
* 39.3.15 isElCapitan(orHigher as boolean = true) as boolean	1160
* 39.3.16 isHighSierra(orHigher as boolean = true) as boolean	1160
* 39.3.17 isLeopard(orHigher as boolean = true) as boolean	1160
* 39.3.18 isLion(orHigher as boolean = true) as boolean	1161
* 39.3.19 isMacOSX as Boolean	1161
* 39.3.20 isMavericks(orHigher as boolean = true) as boolean	1161
* 39.3.21 isMojave(orHigher as boolean = true) as boolean	1161
* 39.3.22 isMonterey(orHigher as boolean = true) as boolean	1162
* 39.3.23 isMountainLion(orHigher as boolean = true) as boolean	1162
* 39.3.24 isSequoia(orHigher as boolean = true) as boolean	1162
* 39.3.25 isSierra(orHigher as boolean = true) as boolean	1163
* 39.3.26 isSnowLeopard(orHigher as boolean = true) as boolean	1163
* 39.3.27 isSonoma(orHigher as boolean = true) as boolean	1163
* 39.3.28 IsTranslated as Integer	1164
* 39.3.29 isVentura(orHigher as boolean = true) as boolean	1164
* 39.3.30 isWindows10(orHigher as boolean = false) as Boolean	1164

* 39.3.31 isWindows11(orHigher as boolean = false) as Boolean	1165
* 39.3.32 isWindows2000(orHigher as boolean = false) as Boolean	1165
* 39.3.33 isWindows7(orHigher as boolean = false) as Boolean	1165
* 39.3.34 isWindows8(orHigher as boolean = false) as Boolean	1166
* 39.3.35 isWindows81(orHigher as boolean = false) as Boolean	1166
* 39.3.36 isWindowsVista(orHigher as boolean = false) as Boolean	1166
* 39.3.37 isWindowsXP(orHigher as boolean = false) as Boolean	1166
* 39.3.38 isYosemite(orHigher as boolean = true) as boolean	1167
* 39.3.39 LogicalRAM as Double	1167
* 39.3.40 MACAddress as string	1167
* 39.3.41 MACAddressString as string	1168
* 39.3.42 MacBoardID as string	1169
* 39.3.43 MacBugFixVersion as Integer	1169
* 39.3.44 MacHasHardwareAcceleratedCoreImage as boolean	1169
* 39.3.45 MachineID(flags as Integer = 15) as string	1170
* 39.3.46 MacMajorVersion as Integer	1171
* 39.3.47 MacMinorVersion as Integer	1171
* 39.3.48 MacModel as string	1172
* 39.3.49 MacROMBootVersion as string	1172
* 39.3.50 MacSerialNumber as string	1172
* 39.3.51 MacUUID as string	1172
* 39.3.52 MacVRAMSize as Int64	1173
* 39.3.53 OSName as string	1173
* 39.3.54 OSVersionString as string	1174
* 39.3.55 PhysicalRAM as Double	1174
* 39.3.56 ProcessorCount(Mode as Integer = 0) as Integer	1175
* 39.3.57 ShortUsername as string	1175
* 39.3.58 SystemFont as string	1175
* 39.3.59 Username as string	1176
* 39.3.60 WinBuildNumber as Integer	1176
* 39.3.61 WinCSDVersion as string	1177
* 39.3.62 WindowsAero as boolean	1177
* 39.3.63 WindowsNativeMachine as Integer	1177
* 39.3.64 WindowsProcessMachine as Integer	1177
* 39.3.65 WinMajorVersion as Integer	1178
* 39.3.66 WinMinorVersion as Integer	1178
* 39.3.67 WinPlatformId as Integer	1178
* 39.3.68 WinProductKey as string	1178
* 39.3.69 WinProductKey(path as string, name as string, keyStartIndex as Integer = 52) as string	1179
* 39.3.70 WinProductType as Integer	1179
* 39.3.71 WinServicePackMajor as Integer	1179
* 39.3.72 WinServicePackMinor as Integer	1180
* 39.3.73 WinSuiteMask as Integer	1180

	159
• 11 Controls	251
– 9.2.1 class TextArea	239
* 9.2.3 SetTextThreadSafeMBS(text as string)	239

• 38 String	1089
– 38.2.1 class TextConverterMBS	1130
* 38.2.3 Canonicalize(name as String) as String	1137
* 38.2.4 Constructor	1137
* 38.2.5 Convert	1137
* 38.2.6 EncodingNames(Mode as Integer = 0) as String()	1138
* 38.2.7 LoadIconvLibrary(path as String, byref Error as String) as boolean	1138
* 38.2.9 ErrorCode as Integer	1138
* 38.2.10 ErrorMessage as String	1138
* 38.2.11 Ignore as Boolean	1139
* 38.2.12 Input as String	1139
* 38.2.13 InputData as MemoryBlock	1139
* 38.2.14 InputEncoding as String	1139
* 38.2.15 InputPosition as Integer	1139
* 38.2.16 LibVersion as Integer	1140
* 38.2.17 Output as String	1140
* 38.2.18 OutputEncoding as String	1140
* 38.2.19 OutputLength as Integer	1140
* 38.2.20 Translit as Boolean	1140
– 38.3.1 class TextEncoding	1142
* 38.3.3 InternetNameMBS as string	1142

	161
• 11 Controls	251
– 11.11.1 class TextField	266
* 11.11.3 SetTextThreadSafeMBS(text as string)	266

• 40 TimeZone	1183
– 40.1.1 class TimeZoneMBS	1183
* 40.1.3 DaylightName as String	1183
* 40.1.4 GmtDeltaHours as Integer	1184
* 40.1.5 GmtDeltaMinutes as Integer	1184
* 40.1.6 GmtDeltaSeconds as Integer	1185
* 40.1.7 GmtDeltaTotalseconds as Integer	1185
* 40.1.8 Latitude as Double	1185
* 40.1.9 Longitude as Double	1186
* 40.1.10 StandardName as String	1186

- 18 Encryption and Hash

345

- ?? Globals

??

- * 18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16 345
- * 18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16 345
- * 18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32 353
- * 18.1.3 CRC16MBS(data as string) as UInt16 346
- * 18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32 346
- * 18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32 347
- * 18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32 347
- * 18.1.7 CRC_32OfStrMBS(s as String) as UInt32 347
- * 18.1.8 CRC_CCITInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32 348
- * 18.1.9 CRC_CCITInMemMBS(address as Ptr, length as Integer) as UInt32 348
- * 18.1.10 CRC_CCITTOfStrContMBS(s as String, prevCRC as UInt32) as UInt32 349
- * 18.1.11 CRC_CCITTOfStrMBS(s as String) as UInt32 349
- * 18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String 349
- * 18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String 349
- * 18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64 349
- * 18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64 352
- * 18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64 350
- * 18.1.21 CurrentUnixTimeMBS as UInt64 353
- * 18.1.22 DecodeFromBase32MBS(data as string) as String 353
- * 18.1.23 EncodeToBase32MBS(data as string) as String 354
- * 18.1.16 GetHash32MBS(s as string) as UInt32 350
- * 18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16 351
- * 18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string 351

• 8 Basic	217
– ?? Globals	??
* 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string	222
* 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock	220
* 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock, len as Integer) as memoryblock	220
* 8.1.10 cloneStringMBS(s as string) as string	220
* 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32	223
* 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
* 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
* 8.1.6 HideCursorMBS	219
* 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color	223
* 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
* 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock, len as Integer) as string	221
* 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
* 8.1.3 ReturnErrPtrMBS as Integer	218
* 8.1.4 ReturnInPtrMBS as Integer	219
* 8.1.5 ReturnOutPtrMBS as Integer	219
* 8.1.7 ShowCursorMBS	219
* 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
* 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222

	165
• 27 MemoryBlock	563
– ?? Globals	??
* 27.1.3 Memoryblock2ptrMBS(mem as memoryblock) as Integer	563
* 27.1.2 NewMemoryBlockFromPtrMBS(ptr as Integer) as memoryblock	563
* 27.1.1 NewMemoryBlockWithBytesMBS(Data as Ptr, size as Integer) as memoryblock	563
* 27.1.4 ptr2MemoryblockMBS(Value as Integer) as memoryblock	564

• 8 Basic	217
– 8.3.1 class UniversalCharacterDetectionMBS	234
* 8.3.3 AddData(data as string)	234
* 8.3.4 Constructor(filter as Integer)	234
* 8.3.5 Finish	234
* 8.3.7 LastCharSet as String	235
* 8.3.9 Report(Charset as string)	235

	167
• 18 Encryption and Hash	345
– 18.2.1 class UUIDMBS	355
* 18.2.3 randomUUID as UUIDMBS	355
* 18.2.4 UUID as UUIDMBS	356
* 18.2.5 Validate(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string	356
* 18.2.6 ValueFormattedString(WithBrackets as Boolean = true) as String	357
* 18.2.7 ValueHexString as String	357
* 18.2.8 ValueMemory as Memoryblock	358
* 18.2.9 ValueString as String	358
* 18.2.11 Lasterror as Integer	358
* 18.2.12 Valid as Boolean	358

• 41 Window	1187
– 41.2.1 class Window	1205
* 41.2.3 ActivateWindowMBS	1205
* 41.2.4 BackingScaleFactorMBS as Double	1206
* 41.2.5 CleanUpTransparentMBS(refValue as Integer)	1206
* 41.2.7 CollapsibleMBS as boolean	1207
* 41.2.8 ConstrainWindowToScreenMBS(animate as boolean)	1207
* 41.2.9 InvalidateRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)	1207
* 41.2.10 InvalidateThreadSafeMBS(EraseBackground as boolean = true)	1208
* 41.2.11 IsFullScreenMBS as Boolean	1208
* 41.2.12 MakeTransparentMBS as Integer	1209
* 41.2.13 RefreshRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)	1209
* 41.2.14 RefreshThreadSafeMBS(EraseBackground as boolean = true)	1210
* 41.2.15 RemoveWindowProxyIconMBS	1210
* 41.2.16 SetTransparencyMBS(value as Integer) as boolean	1211
* 41.2.17 ShowHideToolbarMBS(animate as boolean, value as boolean)	1211
* 41.2.18 SmoothResizeCenteredMBS(width as Integer,height as Integer)	1211
* 41.2.19 SmoothResizeMBS(width as Integer,height as Integer)	1212
* 41.2.20 ToggleFullScreenMBS as Boolean	1213
* 41.2.23 CanBeVisibleWithoutLoginMBS as Boolean	1214
* 41.2.24 collapsedMBS as boolean	1214
* 41.2.25 FullScreenAuxiliaryMBS as Boolean	1214
* 41.2.26 FullScreenPrimaryMBS as Boolean	1215
* 41.2.27 HasborderMBS as boolean	1215
* 41.2.28 HasCaptionMBS as Boolean	1215
* 41.2.29 HasCloseBoxMBS as boolean	1216
* 41.2.30 HasCollapseBoxMBS as boolean	1216
* 41.2.31 HasMaximizeBoxMBS as boolean	1216
* 41.2.32 HasMinimizeBoxMBS as boolean	1217
* 41.2.33 HasNoShadowMBS as boolean	1217
* 41.2.34 HasNoTitleBarMBS as Boolean	1218
* 41.2.35 HasSystemMenuMBS as Boolean	1218
* 41.2.36 HasToolbarButtonMBS as boolean	1219
* 41.2.37 IgnoreClicksMBS as Boolean	1219
* 41.2.38 IsIconicMBS as boolean	1220
* 41.2.39 IsMetalWindowMBS as Boolean	1220
* 41.2.40 IsResizableMBS as Boolean	1220
* 41.2.41 IsZoomedMacMBS as boolean	1221
* 41.2.42 IsZoomedMBS as boolean	1221

* 41.2.43 ModifiedMBS as boolean	1222
* 41.2.44 ToolbarVisibleMBS as boolean	1222
* 41.2.45 TransparencyMBS as single	1222
* 41.2.46 UnifiedTitleAndToolbarMBS as Boolean	1223
* 41.2.47 WindowProxyIconFileMBS as folderitem	1223
* 41.2.48 WinMenuHandleMBS as Integer	1223

- 12 **CoreGraphics** 267
 - 41.2.1 class Window 1205
 - * 41.2.6 ClearTransparencyMBS 1207

	171
• 41 Window	1187
– 41.2.1 class Window	1205
* 41.2.3 ActivateWindowMBS	1205
* 41.2.4 BackingScaleFactorMBS as Double	1206
* 41.2.5 CleanUpTransparentMBS(refValue as Integer)	1206
* 41.2.7 CollapsibleMBS as boolean	1207
* 41.2.8 ConstrainWindowToScreenMBS(animate as boolean)	1207
* 41.2.9 InvalidateRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)	1207
* 41.2.10 InvalidateThreadSafeMBS(EraseBackground as boolean = true)	1208
* 41.2.11 IsFullScreenMBS as Boolean	1208
* 41.2.12 MakeTransparentMBS as Integer	1209
* 41.2.13 RefreshRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)	1209
* 41.2.14 RefreshThreadSafeMBS(EraseBackground as boolean = true)	1210
* 41.2.15 RemoveWindowProxyIconMBS	1210
* 41.2.16 SetTransparencyMBS(value as Integer) as boolean	1211
* 41.2.17 ShowHideToolbarMBS(animate as boolean, value as boolean)	1211
* 41.2.18 SmoothResizeCenteredMBS(width as Integer,height as Integer)	1211
* 41.2.19 SmoothResizeMBS(width as Integer,height as Integer)	1212
* 41.2.20 ToggleFullScreenMBS as Boolean	1213
* 41.2.23 CanBeVisibleWithoutLoginMBS as Boolean	1214
* 41.2.24 collapsedMBS as boolean	1214
* 41.2.25 FullScreenAuxiliaryMBS as Boolean	1214
* 41.2.26 FullScreenPrimaryMBS as Boolean	1215
* 41.2.27 HasborderMBS as boolean	1215
* 41.2.28 HasCaptionMBS as Boolean	1215
* 41.2.29 HasCloseBoxMBS as boolean	1216
* 41.2.30 HasCollapseBoxMBS as boolean	1216
* 41.2.31 HasMaximizeBoxMBS as boolean	1216
* 41.2.32 HasMinimizeBoxMBS as boolean	1217
* 41.2.33 HasNoShadowMBS as boolean	1217
* 41.2.34 HasNoTitleBarMBS as Boolean	1218
* 41.2.35 HasSystemMenuMBS as Boolean	1218
* 41.2.36 HasToolbarButtonMBS as boolean	1219
* 41.2.37 IgnoreClicksMBS as Boolean	1219
* 41.2.38 IsIconicMBS as boolean	1220
* 41.2.39 IsMetalWindowMBS as Boolean	1220
* 41.2.40 IsResizableMBS as Boolean	1220
* 41.2.41 IsZoomedMacMBS as boolean	1221
* 41.2.42 IsZoomedMBS as boolean	1221

* 41.2.43 ModifiedMBS as boolean	1222
* 41.2.44 ToolbarVisibleMBS as boolean	1222
* 41.2.45 TransparencyMBS as single	1222
* 41.2.46 UnifiedTitleAndToolbarMBS as Boolean	1223
* 41.2.47 WindowProxyIconFileMBS as folderitem	1223
* 41.2.48 WinMenuHandleMBS as Integer	1223
– 41.3 Globals	1225
* 41.3.1 MenuBarHeightMBS as Integer	1225

	173
• 42 Windows	1227
– 41.2.1 class Window	1205
* 41.2.21 WinFlashWindowMBS(Invert as boolean)	1213

• 41 Window	1187
– 41.2.1 class Window	1205
* 41.2.3 ActivateWindowMBS	1205
* 41.2.4 BackingScaleFactorMBS as Double	1206
* 41.2.5 CleanUpTransparentMBS(refValue as Integer)	1206
* 41.2.7 CollapsibleMBS as boolean	1207
* 41.2.8 ConstrainWindowToScreenMBS(animate as boolean)	1207
* 41.2.9 InvalidateRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)	1207
* 41.2.10 InvalidateThreadSafeMBS(EraseBackground as boolean = true)	1208
* 41.2.11 IsFullScreenMBS as Boolean	1208
* 41.2.12 MakeTransparentMBS as Integer	1209
* 41.2.13 RefreshRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)	1209
* 41.2.14 RefreshThreadSafeMBS(EraseBackground as boolean = true)	1210
* 41.2.15 RemoveWindowProxyIconMBS	1210
* 41.2.16 SetTransparencyMBS(value as Integer) as boolean	1211
* 41.2.17 ShowHideToolbarMBS(animate as boolean, value as boolean)	1211
* 41.2.18 SmoothResizeCenteredMBS(width as Integer,height as Integer)	1211
* 41.2.19 SmoothResizeMBS(width as Integer,height as Integer)	1212
* 41.2.20 ToggleFullScreenMBS as Boolean	1213
* 41.2.23 CanBeVisibleWithoutLoginMBS as Boolean	1214
* 41.2.24 collapsedMBS as boolean	1214
* 41.2.25 FullScreenAuxiliaryMBS as Boolean	1214
* 41.2.26 FullScreenPrimaryMBS as Boolean	1215
* 41.2.27 HasborderMBS as boolean	1215
* 41.2.28 HasCaptionMBS as Boolean	1215
* 41.2.29 HasCloseBoxMBS as boolean	1216
* 41.2.30 HasCollapseBoxMBS as boolean	1216
* 41.2.31 HasMaximizeBoxMBS as boolean	1216
* 41.2.32 HasMinimizeBoxMBS as boolean	1217
* 41.2.33 HasNoShadowMBS as boolean	1217
* 41.2.34 HasNoTitleBarMBS as Boolean	1218
* 41.2.35 HasSystemMenuMBS as Boolean	1218
* 41.2.36 HasToolbarButtonMBS as boolean	1219
* 41.2.37 IgnoreClicksMBS as Boolean	1219
* 41.2.38 IsIconicMBS as boolean	1220
* 41.2.39 IsMetalWindowMBS as Boolean	1220
* 41.2.40 IsResizableMBS as Boolean	1220
* 41.2.41 IsZoomedMacMBS as boolean	1221
* 41.2.42 IsZoomedMBS as boolean	1221

	175
* 41.2.43 ModifiedMBS as boolean	1222
* 41.2.44 ToolbarVisibleMBS as boolean	1222
* 41.2.45 TransparencyMBS as single	1222
* 41.2.46 UnifiedTitleAndToolbarMBS as Boolean	1223
* 41.2.47 WindowProxyIconFileMBS as folderitem	1223
* 41.2.48 WinMenuHandleMBS as Integer	1223

• 43 XML	1229
– 43.1.1 class XMLValidatorMBS	1229
* 43.1.3 CanonicalXML(XML as String, Mode as Integer = 0) as String	1230
* 43.1.4 Constructor(File as FolderItem)	1230
* 43.1.5 Constructor(XMLSchema as String)	1230
* 43.1.6 Destructor	1231
* 43.1.7 LoadIconvLibrary(path as String, byref Error as String) as boolean	1231
* 43.1.8 Message(index as Integer) as XMLValidatorMessageMBS	1231
* 43.1.9 Messages as XMLValidatorMessageMBS()	1231
* 43.1.10 SetCurrentWorkingDirectory(path as folderitem) as boolean	1231
* 43.1.11 SetCurrentWorkingDirectory(path as String) as boolean	1232
* 43.1.12 ValidateFile(file as FolderItem) as Integer	1232
* 43.1.13 ValidateFile(path as string) as Integer	1233
* 43.1.14 ValidateString(text as string) as Integer	1233
* 43.1.16 MessageCount as Integer	1234
* 43.1.18 Error(message as XMLValidatorMessageMBS)	1234
* 43.1.19 Warning(message as XMLValidatorMessageMBS)	1234
– 43.2.1 class XMLValidatorMessageMBS	1235
* 43.2.3 Constructor	1235
* 43.2.4 Destructor	1235
* 43.2.6 Code as Integer	1235
* 43.2.7 FileName as String	1235
* 43.2.8 IsError as Boolean	1236
* 43.2.9 IsWarning as Boolean	1236
* 43.2.10 Line as Integer	1236
* 43.2.11 Message as String	1236

Chapter 2

List of all classes

• Application	631
• AtomicFlagMBS	195
• AtomicIntegerMBS	198
• AtomicQueueMBS	204
• BlockMBS	299
• ConsoleApplication	652
• CPUIDMBS	269
• DateDifferenceMBS	224
• DeclareCallBackMBS	323
• DeclareFunctionMBS	329
• DeclareLibraryMBS	340
• DelegateCrashExceptionMBS	655
• DesktopApplication	656
• DesktopLabel	253
• DesktopListbox	254
• DesktopProgressbar	255
• DesktopTextArea	237
• DesktopTextField	256
• DesktopUIControl	257

• DesktopWindow	1187
• DirectorySizeMBS	379
• DisplayMBS	703
• DoublePointMBS	242
• DoubleRectMBS	243
• EnvironmentMBS	661
• FileListMBS	397
• FileMappingMBS	369
• FileMapViewMBS	377
• FolderItem	413
• Graphics	459
• HotKeyMBS	463
• IntegerPointMBS	246
• IntegerRectMBS	247
• JSONEntryMBS	467
• JSONIteratorMBS	469
• JSONMBS	472
• Label	259
• LargeBinaryStreamMBS	437
• Listbox	260
• LocaleMBS	286
• MemoryBlock	564
• MemoryBlockMBS	605
• MemoryStorageMBS	611
• MutexMBS	664
• NamedMutexMBS	667
• OpenFileDialogFileTypeMBS	615
• OpenFileDialogItemMBS	617
• OpenFileDialogMBS	619

	179
• ProcessMBS	670
• ProgressBar	261
• RAMStreamMBS	689
• RectControl	263
• RegistrationEngineMBS	697
• ResolutionMBS	712
• ResStreamMBS	448
• SerialPortMBS	715
• ShellMBS	727
• SignalHandlerMBS	1148
• SoftDeclareMBS	306
• Sound	215
• SpamSumMBS	879
• SplineMBS	559
• StdinMBS	449
• StdoutMBS	454
• TextArea	239
• TextConverterMBS	1130
• TextEncoding	1142
• TextField	266
• UniversalCharacterDetectionMBS	234
• UUIDMBS	355
• Window	1205
• XMLValidatorMBS	1229
• XMLValidatorMessageMBS	1235

Chapter 3

List of all modules

• CallDelegateCrashSafeMBS	637
• CallDelegatesMBS	640
• SortMBS	737
• SunTimesMBS	561
• SystemInformationMBS	1155

Chapter 4

List of all global methods

- 39.1.6 AbortMBS 1146
- 26.1.3 ACosHMBS(x as Double) as Double 536
- 26.1.4 ACosMBS(x as Double) as Double 536
- 44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant() 1237
- 26.1.5 ArithmeticShiftMBS(value as UInt64, count as Integer) as UInt64 537
- 44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double() 1237
- 44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64() 1238
- 44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer() 1239
- 44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean 1239
- 44.1.6 ArrayStringMBS(paramArray values as String) as String() 1240
- 44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant() 1240
- 26.1.6 ASinHMBS(x as Double) as Double 537
- 26.1.7 ASinMBS(x as Double) as Double 537
- 26.1.8 ATan2MBS(x as Double, y as Double) as Double 538
- 26.1.9 ATanHMBS(x as Double) as Double 538
- 26.1.10 ATanMBS(x as Double) as Double 539
- 39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string() 1147
- 26.1.11 BitClearMBS(value as UInt64, mask as UInt64) as UInt64 539
- 26.1.12 BitCountMBS(value as UInt64) as Integer 539

- 26.1.13 BitExclMBS(value as UInt64, bitNumber as Integer) as UInt64 540
- 26.1.14 BitInclMBS(value as UInt64, bitNumber as Integer) as UInt64 540
- 26.1.15 BitIsSetMBS(value as UInt64, bitNumber as Integer) as Boolean 540
- 26.1.16 BitValMBS(bitNumber as Integer) as UInt64 541
- 26.1.17 BitwiseDiffMBS(x as UInt64, y as UInt64) as UInt64 541
- 26.1.18 BitwiseNAndMBS(x as UInt64, y as UInt64) as UInt64 541
- 26.1.19 BitwiseNOOrMBS(x as UInt64, y as UInt64) as UInt64 542
- 26.1.20 BitwiseNotMBS(value as UInt64) as UInt64 542
- 26.1.21 BitwiseRotateMBS(value as UInt64, count as Integer, offset as Integer, width as Integer) as UInt64 542
- 8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string 222
- 18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16 345
- 18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16 345
- 18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32 353
- 29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean 643
- 29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean 643
- 29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean 644
- 29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 645
- 29.4.5 CallMethodMBS(target as object, name as string) as boolean 646
- 29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean 646
- 29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean 647
- 29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 647
- 29.4.9 CallMethodOnMainThreadMBS(target as object, name as string) as boolean 648

- 29.4.10 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant) as boolean
649
- 29.4.11 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean
649
- 29.4.12 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean
650
- 14.1.1 CDbLMBS(text as string, byref value as Double, locale as string = "") as boolean
281
- 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string
1098
- 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string
1098
- 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string
1099
- 38.1.14 ClearStringContentMBS(s as String) as Boolean
1100
- 8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock
220
- 8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock, len as Integer) as memoryblock
220
- 8.1.10 cloneStringMBS(s as string) as string
220
- 8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32
223
- 26.1.2 CompareNumbersMBS(v1 as Variant, v2 as Variant) as Integer
535
- 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string
1100
- 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string
1101
- 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string
1101
- 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string
1102
- 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string
1102
- 34.1.1 ConsoleExecuteMBS(path as folderitem, arguments() as string, environment() as string) as Integer
725
- 34.1.2 ConsoleExecuteMBS(path as string, arguments() as string, environment() as string) as Integer
725
- 38.1.20 ContainsWholeWordMBS(Text as String, Word as String) as boolean
1102
- 26.1.22 ConvertFromFloat16MBS(Number as UInt16) as Single
543
- 26.1.23 ConvertToFloat16MBS(Number as Single) as UInt16
543
- 38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string
1126
- 38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string
1127

- 26.1.24 CosHMBS(x as Double) as Double 544
- 26.1.25 CosMBS(x as Double) as Double 545
- 38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer 1104
- 29.4.14 CountProcessesMBS as Integer 651
- 39.1.1 CrashNiceMBS 1143
- 39.1.2 CrashUglyMBS 1143
- 18.1.3 CRC16MBS(data as string) as UInt16 346
- 18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32 346
- 18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32 347
- 18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32 347
- 18.1.7 CRC_32OfStrMBS(s as String) as UInt32 347
- 18.1.8 CRC_CCITInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32 348
- 18.1.9 CRC_CCITInMemMBS(address as Ptr, length as Integer) as UInt32 348
- 18.1.10 CRC_CCITTOfStrContMBS(s as String, prevCRC as UInt32) as UInt32 349
- 18.1.11 CRC_CCITTOfStrMBS(s as String) as UInt32 349
- 18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String 349
- 18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String 349
- 18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64 349
- 18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64 352
- 18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64 350
- 30.2.1 CreateRamStreamMBS(InitialSize as Integer = 0) as RamStreamMBS 696
- 38.1.22 CreateStringMBS(Length as Integer, Content as String) as string 1104
- 26.1.26 CurrencyAddMBS(value1 as Currency, value2 as Currency) as Currency 545
- 26.1.27 CurrencyDivMBS(value1 as Currency, value2 as Integer) as Currency 545
- 26.1.28 CurrencyMulMBS(value1 as Currency, value2 as Integer) as Currency 545
- 26.1.29 CurrencySubMBS(value1 as Currency, value2 as Currency) as Currency 546
- 26.1.30 CurrencyValueMBS(value as string) as Currency 546
- 18.1.21 CurrentUnixTimeMBS as UInt64 353

	187
• 18.1.22 DecodeFromBase32MBS(data as string) as String	353
• 38.1.67 DecodingFromCP1252MBS(s as string) as string	1128
• 38.1.68 DecodingFromHexMBS(s as string) as string	1128
• 38.1.23 DecodingFromHTMLMBS(s as string) as string	1105
• 38.1.69 DecodingFromISO8859MBS(s as string) as string	1128
• 38.1.24 DecodingFromMySQLMBS(s as string) as string	1106
• 38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string	1106
• 38.1.26 DecodingFromURLMBS(s as string) as string	1106
• 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string	1107
• 38.1.28 DecodingFromXMLMBS(s as string) as string	1108
• 39.1.3 DelayMBS(time as Double)	1143
• 39.1.4 DelayMBS(time as Double, mode as Integer)	1144
• 38.1.29 DetectUnicodeMarkersMBS(s as string) as Integer	1108
• 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS	217
• 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS	218
• 32.2.1 DisplayCountMBS as Integer	711
• 26.1.31 DoubleToExtendedStrMBS(x as Double) as string	547
• 26.1.59 DoubleToInt64MBS(value as Double) as Int64	557
• 26.1.60 DoubleToUInt64MBS(value as Double) as UInt64	557
• 38.1.30 EncodeEmailSubjectMBS(s as string) as string	1109
• 18.1.23 EncodeToBase32MBS(data as string) as String	354
• 38.1.8 EncodingNameMBS(extends Text as string) as string	1095
• 38.1.70 EncodingToCP1252MBS(s as string) as string	1129
• 38.1.71 EncodingToHexMBS(s as string) as string	1129
• 38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string	1109
• 38.1.72 EncodingToISO8859MBS(s as string) as string	1130
• 38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string	1110
• 38.1.33 EncodingToURLMBS(s as string) as string	1111
• 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string	1111

- 38.1.35 EncodingToXMLMBS(s as string, options as Integer = 0) as string 1112
- 19.1.1 EndianS16_BtoLMBS(n as Int16) as Int16 359
- 19.1.2 EndianS16_BtoNMBS(n as Int16) as Int16 359
- 19.1.3 EndianS16_LtoBMBS(n as Int16) as Int16 360
- 19.1.4 EndianS16_LtoNMBS(n as Int16) as Int16 360
- 19.1.5 EndianS16_NtoBMBS(n as Int16) as Int16 360
- 19.1.6 EndianS16_NtoLMBS(n as Int16) as Int16 361
- 19.1.7 EndianS32_BtoLMBS(n as Int32) as Int32 361
- 19.1.8 EndianS32_BtoNMBS(n as Int32) as Int32 361
- 19.1.9 EndianS32_LtoBMBS(n as Int32) as Int32 362
- 19.1.10 EndianS32_LtoNMBS(n as Int32) as Int32 362
- 19.1.11 EndianS32_NtoBMBS(n as Int32) as Int32 362
- 19.1.12 EndianS32_NtoLMBS(n as Int32) as Int32 363
- 19.1.13 EndianSwap16MBS(n as UInt16) as UInt16 363
- 19.1.14 EndianSwap32MBS(n as UInt32) as UInt32 363
- 19.1.15 EndianU16_BtoLMBS(n as UInt16) as UInt16 363
- 19.1.16 EndianU16_BtoNMBS(n as UInt16) as UInt16 364
- 19.1.17 EndianU16_LtoBMBS(n as UInt16) as UInt16 364
- 19.1.18 EndianU16_LtoNMBS(n as UInt16) as UInt16 364
- 19.1.19 EndianU16_NtoBMBS(n as UInt16) as UInt16 365
- 19.1.20 EndianU16_NtoLMBS(n as UInt16) as UInt16 365
- 19.1.21 EndianU32_BtoLMBS(n as UInt32) as UInt32 365
- 19.1.22 EndianU32_BtoNMBS(n as UInt32) as UInt32 366
- 19.1.23 EndianU32_LtoBMBS(n as UInt32) as UInt32 366
- 19.1.24 EndianU32_LtoNMBS(n as UInt32) as UInt32 366
- 19.1.25 EndianU32_NtoBMBS(n as UInt32) as UInt32 367
- 19.1.26 EndianU32_NtoLMBS(n as UInt32) as UInt32 367
- 21.2.1 ExchangeFilesMBS(first as folderitem, second as folderitem) as Integer 394
- 39.1.7 ExitMBS(code as Integer) 1146

	189
• 26.1.32 Exp2MBS(x as Double) as Double	547
• 26.1.33 ExpMBS(x as Double) as Double	547
• 26.1.34 ExtendedStrToDoubleMBS(v as string) as Double	548
• 26.1.35 FacMBS(x as Integer) as Double	548
• 26.1.36 FloorMBS(x as Double) as Double	549
• 14.1.2 FormatDateMBS(format as string, value as date, locale as string = "") as string	282
• 14.1.3 FormatDateTimeMBS(format as string, value as dateTime, locale as string = "") as string	283
• 14.1.4 FormatMBS(format as string, value as Double, locale as string = "") as string	283
• 26.1.37 FRExpMBS(inputx as Double, byref expValue as Integer) as Double	549
• 44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr	1242
• 44.1.11 GetDelegateParametersMBS(del as variant) as String	1243
• 44.1.12 GetDelegateTargetMBS(del as variant) as Variant	1243
• 44.1.13 GetDelegateWeakMBS(del as variant) as Boolean	1244
• 32.2.2 GetDisplayMBS(num as Integer) As DisplayMBS	711
• 39.1.10 GetDoubleClickIntervalMBS as Integer	1147
• 44.1.26 GetEncodingOfStringMBS(s as string) as UInt32	1252
• 18.1.16 GetHash32MBS(s as string) as UInt32	350
• 44.1.14 GetObjectMemoryAddressMBS(o as object) as integer	1244
• 44.1.15 GetObjectReferenceCountMBS(o as object) as integer	1245
• 6.1.1 GetSoundMuteMBS as boolean	211
• 6.1.2 GetSoundVolumeLeftMBS as Double	211
• 6.1.3 GetSoundVolumeMBS as Double	211
• 6.1.4 GetSoundVolumeRightMBS as Double	212
• 44.1.16 GetStringMemoryAddressMBS(s as string) as integer	1245
• 44.1.17 GetStringReferenceCountMBS(s as string) as integer	1246
• 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()	1112
• 38.1.37 GetStringsFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113	
• 38.1.38 GetStringsFromDataMBS(data as String, MinLength as Integer = 0) as string()	1113
• 38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string	1114

- 44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant() 1247
- 44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer 1248
- 44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant 1248
- 44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary() 1249
- 44.1.23 GetVariantTypeMBS(va as variant) as Integer 1250
- 38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean 1114
- 38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean 1115
- 38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string 1116
- 8.1.6 HideCursorMBS 219
- 26.1.38 HiWordMBS(i as Integer) as Integer 550
- 26.1.39 HypotMBS(x as Double, y as Double) as Double 550
- 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093
- 38.1.7 InStrBytesMBS(target as string, find as string) as Integer 1094
- 26.1.61 Int64ToDoubleMBS(value as Int64) as Double 558
- 8.1.18 Integer2ColorMBS(intValue as UInt32) as Color 223
- 38.1.43 IsASCIIStringMBS(s as string) as boolean 1117
- 38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean 1117
- 26.1.40 IsFiniteMBS(x as Double) as boolean 550
- 26.1.41 IsInfMBS(x as Double) as boolean 550
- 26.1.42 IsNaNMBS(x as Double) as boolean 551

	191
• 26.1.1 IsValidCreditCardNumberMBS(Number as String) as boolean	535
• 38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double	1117
• 38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double	1118
• 26.1.43 Log10MBS(x as Double) as Double	551
• 26.1.44 Log2MBS(x as Double) as Double	551
• 26.1.45 LogicalShiftMBS(value as UInt64, count as Integer) as UInt64	552
• 26.1.46 LogMBS(x as Double) as Double	552
• 26.1.47 LoWordMBS(i as Integer) as Integer	552
• 10.1.1 MakeDoublePointMBS(x as Double, y as Double) as DoublePointMBS	241
• 10.1.2 MakeDoubleRectMBS(left as Double, top as Double, width as Double, height as Double) as DoubleRectMBS	241
• 10.1.3 MakeIntegerPointMBS(x as Integer, y as Integer) as IntegerPointMBS	241
• 10.1.4 MakeIntegerRectMBS(left as Integer, top as Integer, width as Integer, height as Integer) as IntegerRectMBS	242
• 27.1.3 Memoryblock2ptrMBS(mem as memoryblock) as Integer	563
• 8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string	221
• 8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock, len as Integer) as string	221
• 41.3.1 MenuBarHeightMBS as Integer	1225
• 39.1.9 MillisecondsMBS as Double	1147
• 18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16	351
• 38.1.47 NativeStringMBS(s as string) as string	1119
• 27.1.2 NewMemoryBlockFromPtrMBS(ptr as Integer) as memoryblock	563
• 27.1.1 NewMemoryBlockWithBytesMBS(Data as Ptr, size as Integer) as memoryblock	563
• 26.1.48 NormInvMBS(p as Double, mu as double = 0.0, sigma as double = 1.0) as double	553
• 44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean	1250
• 8.1.13 OSTypeFromStringMBS(str as string) as Integer	221
• 14.1.5 ParseDateMBS(format as string, text as string, byref value as date, locale as string = "") as boolean	284
• 14.1.6 ParseDateTimeMBS(format as string, text as string, byref value as dateTime, locale as string = "") as boolean	286
• 26.1.49 PowMBS(x as Double, y as Double) as Double	553

- 27.1.4 ptr2MemoryblockMBS(Value as Integer) as memoryblock 564
- 38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string 1119
- 26.1.50 RandomExponentialDistributionMBS(lambda as Double) as double 554
- 26.1.51 RandomNormalDistributionMBS(Mean as Double, StdDev as Double) as double 554
- 26.1.52 RandomPoissonDistributionMBS(Mean as Integer) as Integer 554
- 38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string 1095
- 38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string 1120
- 38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string 1120
- 38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String 1121
- 38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string 1121
- 26.1.53 RoundMBS(x as Double, decimals as Integer = 0) as Double 555
- 38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string 1122
- 21.2.7 SetCurrentWorkingDirectoryMBS(path as folderitem) as boolean 395
- 21.2.8 SetDllDirectoryMBS(path as folderitem) as boolean 396
- 44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32) 1253
- 6.1.5 SetSoundMuteMBS(mute as boolean) 212
- 6.1.6 SetSoundVolumeLeftMBS(percent as Double) 212
- 6.1.7 SetSoundVolumeMBS(percent as Double) 212
- 6.1.8 SetSoundVolumeRightMBS(percent as Double) 213
- 29.4.13 SetThreadNameMBS(name as string) 651
- 44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant) 1251
- 8.1.7 ShowCursorMBS 219
- 11.1.2 ShowModalThreadSafeMBS(extends theMessageDialog as MessageDialog) 251
- 11.1.1 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as DesktopWindow) 251
- 11.1.3 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as window) 252
- 26.1.54 SinHMBS(x as Double) as Double 555
- 26.1.55 SinMBS(x as Double) as Double 556

	193
• 39.1.5 SleepMBS(time as Double)	1145
• 38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string()	1096
• 38.1.54 SQLReplaceBooleanMBS(SQL as string) as string	1122
• 26.1.56 SqrtMBS(x as Double, y as Double) as Double	556
• 38.1.55 StrCompBytesMBS(a as string, b as string) as Integer	1122
• 38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer	1123
• 38.1.57 StringANDMBS(a as string,b as string) as string	1123
• 8.1.14 StringFromOSTypeMBS(value as Integer) as string	222
• 38.1.58 StringIsHTMLreadyMBS(s as string) as boolean	1124
• 38.1.59 StringIsXMLreadyMBS(s as string) as boolean	1124
• 38.1.60 StringORMBS(a as string,b as string) as string	1124
• 8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock	222
• 38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
• 38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string	1125
• 38.1.63 StrMBS(d as Double) as string	1126
• 26.1.57 TanHMBS(x as Double) as Double	556
• 26.1.58 TanMBS(x as Double) as Double	557
• 26.1.62 UInt64ToDoubleMBS(value as UInt64) as Double	558
• 38.1.64 UnicodeStringMBS(s as string) as string	1126
• 32.2.4 UpdateDisplayCountMBS	712
• 18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string	351
• 21.2.9 VolumeFreeSizePathMBS(Path as String) as Int64	397
• 21.2.10 VolumeSizePathMBS(Path as String) as Int64	397

Chapter 5

Atomic

5.1 class AtomicFlagMBS

5.1.1 class AtomicFlagMBS

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for an atomic flag.

Example:

```
Var f As New AtomicFlagMBS

// lock
Var b As Boolean = f.TestAndSet
If Not b Then
// got lock
Break

// unlock
f.Clear
Else
// failed to lock
Break
End If
```

Notes: This is an lock-free atomic boolean type.
Can be used to make your own semaphore or critical section class.

See also AtomicIntegerMBS class.

Blog Entries

- [MBS Xojo Plugins, version 25.2pr5](#)

5.1.2 Methods

5.1.3 Clear

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Clears the flag.

5.1.4 Operator_Compare(other as AtomicFlagMBS) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares two atomic flags.

Example:

```
Var a As AtomicFlagMBS = True
Var b As AtomicFlagMBS = False
```

```
If a > b Then
  // works
  Break
Else
  // failed
  Break
End If
```

5.1.5 Operator_Convert as Boolean

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Converts the flag to a boolean.

See also:

- [5.1.6 Operator_Convert\(value as Boolean\)](#)

196

5.1.6 Operator_Convert(value as Boolean)

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Converts a boolean to be the new value.

See also:

- 5.1.5 Operator_Convert as Boolean

5.1.7 Test as Boolean

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks current value and reports back.

5.1.8 TestAndSet as Boolean

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Tests value and sets flag.

Example:

```
Var f As New AtomicFlagMBS

// lock
Var b As Boolean = f.TestAndSet
If Not b Then
// got lock
Break

// unlock
f.Clear
Else
// failed to lock
Break
End If
```

Notes: Atomically changes the state of a flag to set (true) and returns the value it held before.

5.1.9 Properties

5.1.10 Value as Boolean

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The current value.

Notes: (Read and Write property)

5.2 class AtomicIntegerMBS

5.2.1 class AtomicIntegerMBS

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for an atomic integer.

Example:

```
// this should be a property
Var a As New AtomicIntegerMBS

// try to get new unique ID
Do
  Var Expected As Int64 = a
  Var newCount As Int64 = Expected+1
  Var r As Boolean = a.CompareExchange(Expected, newCount)

  If r Then
    // We got our new ID, so continue
    Exit
  Else
    // some other thread was quicker, so try again!
    continue
  End If
Loop
```

Notes: This class is to have one object as counter or flag to get/set from multiple threads.

See also AtomicFlagMBS class.

Blog Entries

- [MBS Xojo Plugins, version 25.2pr5](#)

5.2.2 Methods

5.2.3 Add(Value as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds an integer.

Notes: Atomically replaces the current value with the result of arithmetic addition of the value and argument.

Returns the value immediately preceding the effects of this function.

5.2.4 BitwiseAnd(Value as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Performs a thread safe bitwise and operation.

Notes: Atomically replaces the current value with the result of bitwise AND of the value and argument.

Returns the value immediately preceding the effects of this function.

5.2.5 BitwiseOr(Value as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Performs a thread safe bitwise or operation.

Example:

```
Var a As New AtomicIntegerMBS
```

```
// put in some value  
a = 2
```

```
Var x As Int64 = a.BitwiseOr(4)
```

```
// later read the value  
Var Value As Int64 = a
```

```
// Value is 2 or 4 = 6
```

```
Break
```

Notes: Atomically replaces the current value with the result of bitwise OR of the value and argument.

Returns the value immediately preceding the effects of this function.

5.2.6 BitwiseXOr(Value as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Performs a thread safe bitwise xor operation.

Notes: Atomically replaces the current value with the result of bitwise XOR of the value and argument.

Returns the value immediately preceding the effects of this function.

5.2.7 CompareExchange(byref expected as Int64, desired as Int64) as Boolean

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares value and sets to new value.

Example:

```
Var a As New AtomicIntegerMBS

// put in some value
a = 3

// we check if value is 4, we set to 5
Var Expected1 As Int64 = 4
Var r1 As Boolean = a.CompareExchange(Expected1, 5)

// we check if value is 3, we set to 6
Var Expected2 As Int64 = 3
Var r2 As Boolean = a.CompareExchange(Expected2, 6)

// later read the value
Var Value As Int64 = a

Break
```

Notes: On success returns true.

On failure returns false and assigns current value to expected parameter.

5.2.8 Decrement as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Decrements the value.

Notes: Returns new value.

5.2.9 Exchange(Value as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Exchanges current value with new value.

Example:


```
Var a As New AtomicIntegerMBS

// put in some value
a = 3

// later read the value
Var oldValue As Int64 = a.Exchange(5)
Var NewValue As Int64 = a

Break
```

Notes: Returns old value.

5.2.10 Increment as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Increments the value.

Example:

```
Var a As New AtomicIntegerMBS

// put in some value
a.Value = 3

// increment it from multiple threads
Call a.Increment

// later read the value
Var v As Int64 = a.Value
MessageBox v.ToString
```

Notes: Returns new value.

5.2.11 Operator_Add(other as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds a value.

5.2.12 Operator_Compare(other as AtomicIntegerMBS) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares two atomic integers.

Example:

```
Var a As AtomicIntegerMBS = 123
Var b As AtomicIntegerMBS = 234
```

```
If a < b Then
  // works
  Break
Else
  // failed
  Break
End If
```

```
If a > b Then
  // failed
  Break
Else
  // works
  Break
End If
```

5.2.13 Operator_Convert as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Converts value to integer.

Example:

```
Var a As New AtomicIntegerMBS
```

```
// put in some value
a = 3
```

```
// later read the value
Var v As Int64 = a
MessageBox v.ToString
```

See also:

- 5.2.14 Operator_Convert(value as Int64)

5.2.14 Operator_Convert(value as Int64)

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Converts value on assignment.

See also:

- 5.2.13 Operator_Convert as Int64

202

5.2.15 Operator_Subtract(other as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Subtracts a value.

5.2.16 Subtract(Value as Int64) as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Subtracts a value.

Notes: Atomically replaces the current value with the result of arithmetic addition of the value and argument.

Returns the value immediately preceding the effects of this function.

5.2.17 Properties

5.2.18 Value as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get or set current value.

Notes: Atomically reads or replaces the current value.

(Read and Write property)

5.3 class AtomicQueueMBS

5.3.1 class AtomicQueueMBS

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for an atomic queue.

Example:

```
Var f As New AtomicQueueMBS

// one can add jobs
f.Push New MyJob
f.Push New MyJob

// someone else can pop them from the queue
Do
Var job As MyJob = f.PopLast // or f.PopFirst

job.Process

Loop Until f.count = 0
```

Notes: This can be used for e.g. a job queue, where multiple threads add or remove jobs.

Blog Entries

- [MBS Xojo Plugins, version 25.2pr6](#)
- [MBS Xojo Plugins, version 25.2pr5](#)

5.3.2 Methods

5.3.3 Add(values as AtomicQueueMBS)

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the values of the other queue to this queue.

Example:

```
Var a As New AtomicQueueMBS

a.Push 123
a.Push 234

Var b As New AtomicQueueMBS
b.Add a
```

5.3. CLASS ATOMICQUEUEMBS

205

```
// this uses Operator_Convert  
Var values() As Variant = b
```

Break

See also:

- 5.3.4 Add(values() as Variant)

205

5.3.4 Add(values() as Variant)

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the given values to the queue.

Example:

```
Var a As New AtomicQueueMBS
```

```
Var newValues() As Variant  
newValues.Add 123  
newValues.Add 234
```

a.Add newValues

```
Var values() As Variant = a  
// check in debugger
```

Break

Notes: Like if you push each of the value in the array.

See also:

- 5.3.3 Add(values as AtomicQueueMBS)

204

5.3.5 Clear

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Clears the list.

Example:

```
Var f As New AtomicQueueMBS
```

```
// put a value on the queue  
f.Push 123  
Var c As Integer = f.Count
```

```
// clear it
f.Clear

MessageBox c.ToString + " ->" + f.Count.ToString
```

5.3.6 Insert(value as Variant)

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Inserts a new value on bottom of the queue.

Notes: Inserts value at position 0 in the array.

5.3.7 Operator__Add(value as Variant) as AtomicQueueMBS

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds a value to the queue.

Example:

```
Var a As New AtomicQueueMBS
```

```
a = a + 123
```

```
a = a + 234
```

```
Var values() As Variant = a
```

```
Break
```

5.3.8 Operator__Compare(other as AtomicQueueMBS) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares two queues.

Notes: Two queues are equal if

- you compare same object against itself
- both have same sizes and each of the variant pointers are the same.

This allows you to compare a copy of the queue or one with the same objects.

5.3.9 Operator_Convert as Variant()

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Converts the queue to an array.

Example:

```

Var q As New AtomicQueueMBS

q.Push New MyJob
q.Push New MyJob
q.Push New MyJob
q.Push New MyJob
q.Push New MyJob

Var other As New AtomicQueueMBS
other.add(q)

If other = q Then
Break // equal
Else
Var qv() As Variant = q.Values
Var ov() As Variant = other.values
// not equal, so check arrays in debugger

Break
End If

```

Notes: Same as calling values function.

5.3.10 PopFirst as Variant

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns the first item and removes it from the queue.

Example:

```

Var q As New AtomicQueueMBS

q.Push 1
q.Push 2
q.Push 3
q.Push 4
q.Push 5

Var f As Variant = q.PopFirst // 1
Var l As Variant = q.PopLast // 5

```

```
Var c As Integer = q.Count // 3 remaining
```

```
Break
```

Notes: This is for a first-in-first-out queue.

5.3.11 PopLast as Variant

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns the last item and removes it from the queue.

Notes: This is for a first-in-last-out queue, also known as a stack.

5.3.12 Push(value as Variant)

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Pushes a new value on top of the queue.

Notes: Appends value to the array.

5.3.13 Values as Variant()

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries array of all values.

5.3.14 Properties

5.3.15 Count as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries the number of values.

Example:

```
Var f As New AtomicQueueMBS
```

```
f.Push 123
```

```
MessageBox f.Count.ToString
```

```
f.Push 234
```


MessageBox f.Count.ToString

Notes: (Read only property)

5.3.16 TopValue as Variant

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Looks up the top value.

Notes: Does not remove the top value, but keeps it in the queue.
(Read and Write property)

Chapter 6

Audio

6.1 Globals

6.1.1 GetSoundMuteMBS as boolean

Plugin Version: 6.2, Platforms: macOS, Windows, Targets: All.

Function: Checks the mute state of the sound output.

Example:

```
if GetSoundMuteMBS then
  Title="on"
else
  Title="off"
end if
```

Notes: Returns false on any error.

6.1.2 GetSoundVolumeLeftMBS as Double

Plugin Version: 7.1, Platforms: macOS, Windows, Targets: All.

Function: Gets the current volume for the left channel.

Notes: Same as GetSoundVolumeMBS, but only for the left channel.

6.1.3 GetSoundVolumeMBS as Double

Plugin Version: 5.0, Platforms: macOS, Windows, iOS, Targets: All.

Function: Gets the current volume.

Example:

```
Var d as Double
```

```
d=GetSoundVolumeMBS
```

Notes: Volume is from 0 to 1.0

Uses QuickTime on Mac OS Classic, CoreAudio on Mac OS X and the Mixer on Windows. Added iOS support with version 24.5, but it requires an active audio session to get volume.

6.1.4 GetSoundVolumeRightMBS as Double

Plugin Version: 7.1, Platforms: macOS, Windows, Targets: All.

Function: Gets the current volume for the right channel.

Notes: Same as GetSoundVolumeMBS, but only for the right channel.

6.1.5 SetSoundMuteMBS(mute as boolean)

Plugin Version: 6.2, Platforms: macOS, Windows, Targets: All.

Function: Sets sound mute to the give state.

Example:

```
SetSoundMuteMBS true // set mute on
```

6.1.6 SetSoundVolumeLeftMBS(percent as Double)

Plugin Version: 7.1, Platforms: macOS, Windows, Targets: All.

Function: Sets the volume on the left channel.

Notes: Volume is from 0 to 1.0

Uses QuickTime on Mac OS Classic, CoreAudio on Mac OS X and the Mixer on Windows.

6.1.7 SetSoundVolumeMBS(percent as Double)

Plugin Version: 5.0, Platforms: macOS, Windows, Targets: All.

Function: Sets the volume.

Example:

SetSoundVolumeMBS 1.0

Notes: Volume is from 0 to 1.0

Uses QuickTime on Mac OS Classic, CoreAudio on Mac OS X and the Mixer on Windows.

6.1.8 SetSoundVolumeRightMBS(percent as Double)

Plugin Version: 7.1, Platforms: macOS, Windows, Targets: All.

Function: Sets the volume on the right channel.

Notes: Volume is from 0 to 1.0

Uses QuickTime on Mac OS Classic, CoreAudio on Mac OS X and the Mixer on Windows.

Chapter 7

AVFoundation

7.1 class Sound

7.1.1 class Sound

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in sound class in Xojo.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.mp4")
Var s as sound = f.OpenAsSound
Var a as AVAudioPlayerMBS = s.AVAudioPlayerMBS
MsgBox str(a.duration)+" seconds long"
call a.play
```


Chapter 8

Basic

8.1 Globals

8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Calculates the difference between two dates.

Example:

```
Var d as date
Var e as date
Var r as DateDifferenceMBS
Var s as string
```

```
d=new date
d.Year=2008
d.Month=7
d.Day=2
d.Hour=10
d.Minute=48
d.Second=22
```

```
e=new date
e.Year=2010
e.Month=7
e.Day=1
e.Hour=10
e.Minute=36
e.Second=0
```

```
r=d.DifferenceMBS(e)
```

```
s= "Years: "+str(r.Year)+EndOfLine
s=s+"Months: "+str(r.month)+EndOfLine
s=s+"Days: "+str(r.day)+EndOfLine
s=s+"Hours: "+str(r.hour)+EndOfLine
s=s+"Minutes: "+str(r.Minute)+EndOfLine
s=s+"Seconds: "+str(r.Second)+EndOfLine
```

```
MsgBox s
```

```
// shows: "Years: 1 Months: 11 Days: 29 Hours: 22 Minutes: 47 Seconds: 38"
```

Notes: Returns nil on any errors. (one of the dates is nil or property getter don't work)
Valid only for dates from the gregorian calendar.

Calculates the difference between the older and newer date. The dates are sorted, so the difference is always a positive.

You can see Swap property to see if first date is after second date.

See also:

- 8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS
218

8.1.2 DifferenceMBS(extends StartDate as dateTime, EndDate as dateTime) as DateDifferenceMBS

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the difference between two dates.

Notes: Returns nil on any errors. (one of the dates is nil or property getter don't work)

Valid only for dates from the gregorian calendar.

Calculates the difference between the older and newer date. The dates are sorted, so the difference is always a positive.

You can see Swap property to see if first date is after second date.

Please make sure both dates are in the same timezone.

See also:

- 8.1.1 DifferenceMBS(extends StartDate as date, EndDate as date) as DateDifferenceMBS 217

8.1.3 ReturnErrPtrMBS as Integer

Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** A callback to use for GS8.
Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

8.1.4 ReturnIntPtrMBS as Integer

Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** A callback to use for GS8.
Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

8.1.5 ReturnOutPtrMBS as Integer

Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** A callback to use for GS8.
Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

8.1.6 HideCursorMBS

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Hides the mouse Cursor.

Example:

```
HideCursorMBS
```

8.1.7 ShowCursorMBS

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Shows the mouse Cursor.

Example:

```
ShowCursorMBS
```

8.1.8 cloneMemoryBlockMBS(s as memoryblock) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Clones the memoryblock.

Example:

```
Var m as MemoryBlock = "Hello"  
Var n as MemoryBlock = cloneMemoryBlockMBS(m)
```

```
m.Byte(1)=asc("a")  
n.Byte(1)=asc("u")
```

```
Var a as string = m  
Var b as string = n
```

```
MsgBox a+" "+b
```

Notes: May return nil on low memory conditions.

8.1.9 cloneMemoryBlockWithLengthMBS(s as memoryblock,len as Integer) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Clones the memoryblock.

Example:

```
Var m as MemoryBlock = "Hello"  
Var n as MemoryBlock = cloneMemoryBlockWithLengthMBS(m,5)
```

```
m.Byte(1)=asc("a")  
n.Byte(1)=asc("u")
```

```
Var a as string = m  
Var b as string = n
```

```
MsgBox a+" "+b
```

Notes: May return nil on low memory conditions.

8.1.10 cloneStringMBS(s as string) as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: Clones the string.

Notes: May return "" on low memory conditions.

The encoding of the string is copied in RB 4.5 or newer.

8.1.11 MemoryBlockToStringMBS(s as memoryblock) as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a copy of the content of the memoryblock as a string.

Example:

```
Var m as MemoryBlock = NewMemoryBlock(6)
```

```
m.CString(0)="Hello"
```

```
MsgBox MemoryBlockToStringMBS(m)
```

Notes: May return "" on low memory conditions.

As some memoryblocks don't have a known length, you can provide one as a second parameter.

8.1.12 MemoryBlockToStringWithLengthMBS(s as memoryblock,len as Integer) as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a copy of the content of the memoryblock as a string.

Example:

```
Var m as MemoryBlock = NewMemoryBlock(6)
```

```
m.CString(0)="Hello"
```

```
MsgBox MemoryBlockToStringWithLengthMBS(m,5)
```

Notes: May return "" on low memory conditions.

As some memoryblocks don't have a known length, you can provide one as a second parameter.

8.1.13 OStringFromMemoryBlockMBS(str as string) as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns an integer representing a 4 byte String.

Example:

```
MsgBox str(OSTypeFromStringMBS("MBSP"))
```

Notes: This OSType datatype is used on Mac OS for 4 letter codes.
e.g. the type and creator code for a file.

Blog Entries

- [Write audio file with samples using AVFoundation](#)

8.1.14 StringFromOSTypeMBS(value as Integer) as string

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a 4 byte string for the given integer.

Example:

```
MsgBox StringFromOSTypeMBS(1296192336) // shows MBSP
```

Notes: This OSType datatype is used on Mac OS for 4 letter codes.
e.g. the type and creator code for a file.

8.1.15 StringToMemoryBlockMBS(s as string) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a copy of the content of the string as a memoryblock.

Example:

```
Var m as MemoryBlock = StringToMemoryBlockMBS("Hello")
```

```
MsgBox m.StringValue(0,5)
```

Notes: May return nil on low memory conditions.

8.1.16 BitwiseXORStringBytesMBS(s as string, v as Integer) as string

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Bitwise XOR with a string's bytes.

Example:

```
Var a as string = "Hello"
Var b as string = BitwiseXORStringBytesMBS(a,1) // encode
```

```

Var c as string = BitwiseXORStringBytesMBS(b,1) // decode
MsgBox b
MsgBox c

```

Notes: Memory usage is around $\text{lenb}(s)*2$.

Returns nil on low memory.

v should be in range of 0 to 255.

If you use this function two times on a string, you get the original back.

Any encoding information is lost.

Blog Entries

- [MBS Xojo Plugins, version 17.3pr1](#)

8.1.17 Color2IntegerMBS(colorValue as Color) as UInt32

Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Returns the RGB value of the color as an integer.

Example:

```

Var c as color
Var i as UInt32

```

```

c=rgb(255,255,255)
i=Color2IntegerMBS(c)

```

```

MsgBox hex(i) // FFFFFFFF

```

```

c=rgb(&h12,&H34,&h56)
i=Color2IntegerMBS(c)

```

```

MsgBox hex(i) // 123456

```

Notes: Same as:

```

i=c.red*65536+c.green*256+c.blue

```

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr14](#)

8.1.18 Integer2ColorMBS(intValue as UInt32) as Color

Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Returns the integer as if it is a color.

Example:

```

Var c as color
Var i as UInt32
c=Integer2ColorMBS(i)

```

Notes: Same as:

```
c=rgb(i\65536 mod 256, i\256 mod 256, i mod 256)
```

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr14](#)

8.2 class DateDifferenceMBS

8.2.1 class DateDifferenceMBS

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class to calculate the difference between two dates.

Example:

```
// calculate difference between now and a date in 2008.
```

```

Var d as date
Var r as DateDifferenceMBS
Var s as string
Var c as Clipboard

d=new date
d.Year=2008
d.Month=7
d.Day=2
d.Hour=10
d.Minute=48
d.Second=22

Var e as new date // today

r=new DateDifferenceMBS(d, e)

s= "Years: " +str(r.Year)+EndOfLine
s=s+ "Months: " +str(r.month)+EndOfLine
s=s+ "Days: " +str(r.day)+EndOfLine
s=s+ "Hours: " +str(r.hour)+EndOfLine
s=s+ "Minutes: " +str(r.Minute)+EndOfLine
s=s+ "Seconds: " +str(r.Second)+EndOfLine

MsgBox s

```


Notes: Calculates the difference between the older and newer date. The dates are sorted, so the difference is always a positive.

You can see Swap property to see if first date is after second date.

This class allows you to know exactly how many months/years are between two dates. Something that TotalSeconds calculations will not give you!

Blog Entries

- [MBS Xojo Plugins, version 25.1pr5](#)
- [MBS Xojo Plugins, version 24.1pr3](#)
- [Cleanup Xojo Plugins](#)
- [MBS Xojo Plugins, version 20.3pr5](#)
- [MBS Xojo Plugins, version 19.5pr4](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr6](#)
- [MBS Real Studio Plugins, version 12.4pr9](#)
- [MonkeyBread Software Releases the MBS Plugins 8.3](#)

8.2.2 Methods

8.2.3 Calc(StartDate as date, EndDate as date) as boolean

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Calculates the difference between two dates.

Example:

```
Var d as date
Var e as date
Var r as DateDifferenceMBS
Var s as string
Var c as Clipboard
```

```
d=new date
d.Year=2008
d.Month=7
d.Day=2
d.Hour=10
d.Minute=48
d.Second=22
```

```
e=new date
```

```
e.Year=2010
e.Month=7
e.Day=1
e.Hour=10
e.Minute=36
e.Second=0
```

```
r=new DateDifferenceMBS
```

```
if r.Calc(d,e) then
```

```
s= "Years: "+str(r.Year)+EndOfLine
s=s+"Months: "+str(r.month)+EndOfLine
s=s+"Days: "+str(r.day)+EndOfLine
s=s+"Hours: "+str(r.hour)+EndOfLine
s=s+"Minutes: "+str(r.Minute)+EndOfLine
s=s+"Seconds: "+str(r.Second)+EndOfLine
```

```
MsgBox s
```

```
// shows: "Years: 1 Months: 11 Days: 29 Hours: 22 Minutes: 47 Seconds: 38"
end if
```

Notes: Returns false on any errors and true on success.
Valid only for dates from the gregorian calendar.
Sets ready property.

Calculates the difference between the older and newer date. The dates are sorted, so the difference is always a positive.

You can see Swap property to see if first date is after second date.

See also:

- 8.2.4 Calc(StartDate as dateTime, EndDate as dateTime) as boolean

226

8.2.4 Calc(StartDate as dateTime, EndDate as dateTime) as boolean

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the difference between two dates.

Notes: Returns false on any errors and true on success.
Valid only for dates from the gregorian calendar.
Sets ready property.

Calculates the difference between the older and newer date. The dates are sorted, so the difference is always a positive.

8.2. CLASS DATEDIFFERENCEMBS 227

You can see Swap property to see if first date is after second date.

Please make sure both dates are in the same timezone.

See also:

- 8.2.3 Calc(StartDate as date, EndDate as date) as boolean 225

8.2.5 Constructor

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The dummy constructor doing nothing.

See also:

- 8.2.6 Constructor(StartDate as date, EndDate as date) 227
- 8.2.7 Constructor(StartDate as dateTime, EndDate as dateTime) 227

8.2.6 Constructor(StartDate as date, EndDate as date)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Calculates the difference between two dates.

Notes: Internally calls Calc and sets ready property.

See also:

- 8.2.5 Constructor 227
- 8.2.7 Constructor(StartDate as dateTime, EndDate as dateTime) 227

8.2.7 Constructor(StartDate as dateTime, EndDate as dateTime)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the difference between two dates.

Notes: Internally calls Calc and sets ready property.

Please make sure both dates are in the same timezone.

See also:

- 8.2.5 Constructor 227
- 8.2.6 Constructor(StartDate as date, EndDate as date) 227

8.2.8 isLeapYear(year as Integer) as boolean

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether a year is a leap year.

Example:

```
if DateDifferenceMBS.isLeapYear(2004) then
msgbox "2004 is a leap year"
else
msgbox "2004 is not a leap year"
end if
```

Notes: Valid only for dates from the gregorian calendar.

8.2.9 Properties

8.2.10 Day as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The day of the date difference.

Notes: (Read only property)

8.2.11 EndDate as Variant

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The end date used to do the calculation.

Notes: Value can be DateTime or Date depending of method used.
(Read only property)

8.2.12 EndDay as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The day of the end date.

Notes: (Read only property)

8.2.13 EndHour as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The hour of the end date.

Notes: (Read only property)

8.2.14 EndMinute as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The minute of the end date.

Notes: (Read only property)

8.2.15 EndMonth as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The month of the end date.

Notes: (Read only property)

8.2.16 EndSecond as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The second of the end date.

Notes: (Read only property)

8.2.17 EndTotalSeconds as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total seconds value of the end date.

Notes: (Read and Write property)

8.2.18 EndYear as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The year of the end date.

Notes: (Read only property)

8.2.19 Hour as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The hour of the date difference.

Notes: (Read only property)

8.2.20 Minute as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The minute of the date difference.

Notes: (Read only property)

8.2.21 Month as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The month of the date difference.

Notes: (Read only property)

8.2.22 Ready as Boolean

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the date calculation has been performed.

Notes: Using the constructor with nil dates can lead into an invalid state in which ready is false.
(Read only property)

8.2.23 Second as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The second of the date difference.

Notes: (Read only property)

8.2.24 StartDate as Variant

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The start date of the date used.

Notes: Value can be DateTime or Date depending of method used.
(Read only property)

8.2.25 StartDay as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The day of the start date.

Notes: (Read only property)

8.2.26 StartHour as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The hour of the start date.

Notes: (Read only property)

8.2.27 StartMinute as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The minute of the start date.

Notes: (Read only property)

8.2.28 StartMonth as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The month of the start date.

Notes: (Read only property)

8.2.29 StartSecond as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The second of the start date.

Notes: (Read only property)

8.2.30 StartTotalSeconds as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total seconds value of the start date.

Notes: (Read and Write property)

8.2.31 StartYear as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The year of the start date.

Notes: (Read only property)

8.2.32 Swap as Boolean

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether dates have been swapped.

Notes: Dates are swapped if enddate is before startdate.

(Read only property)

8.2.33 TotalDay as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of days between

Example:

```
Var d as new date(2011, 9, 25)
```

```
Var e as new date(2012, 1, 1)
```

```
Var diff as new DateDifferenceMBS
```

```
if diff.Calc(d, e) then
```

```
MsgBox str(diff.TotalDay)+" days between dates." // should be 98
```

```
end if
```


Notes: Our day, month and year properties are about how many days, months and years are between start and end dates.

This property gives you the total number of days.

So day may be 6 and month be 3 and totaldays could be 98 if there are two months with 31 days included and one with 30 days.

(Read only property)

8.2.34 TotalSeconds as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total seconds value of the difference between the dates.

Notes: (Read and Write property)

8.2.35 Year as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The year of the date difference.

Notes: (Read only property)

8.3 class UniversalCharacterDetectionMBS

8.3.1 class UniversalCharacterDetectionMBS

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for unival character encoding detection.

Notes: Based on the Mozilla project.

You can pass a string with bytes to this class and receive an encoding as answer. Similar to the Guess-JapaneseEncoding function built into Xojo.

Possible encoding names used by this class: Big5, EUC-JP, EUC-KR, x-euc-tw, gb18030, windows-1252, UTF-8, UTF-16BE, X-ISO-10646-UCS-4-3412, UTF-32BE, X-ISO-10646-UCS-4-2143, UTF-32LE, UTF-16LE.

Blog Entries

- [MBS Xojo Plugins, version 22.5pr3](#)
- [Cleanup Xojo Plugins](#)
- [MBS REALbasic plug-ins version 10.0](#)

8.3.2 Methods

8.3.3 AddData(data as string)

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Passes more data to the engine for analysis.

Notes: The string encoding from Xojo is ignored. Only the raw data bytes are examined.

8.3.4 Constructor(filter as Integer)

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Notes: filter: which languages to detect or ignore. Use the filter* constants.

8.3.5 Finish

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Notify the engine that data no more data will come.

Notes: This method can fire the Report event.

8.3.6 Properties

8.3.7 LastCharSet as String

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last character set reported in the report event.

Notes: (Read and Write property)

8.3.8 Events

8.3.9 Report(Charset as string)

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event called when an encoding was detected.

Notes: This event allows you to pass more and more data and as soon as the event fires you can decide to stop probing.

8.3.10 Constants

Constants

Constant	Value	Description
FilterAll	31	One of the language filter constants. All languages.
FilterChinese	3	One of the language filter constants. Traditional and Simplified Chinese together.
FilterChineseSimplified	1	One of the language filter constants. Simplified Chinese.
FilterChineseTraditional	2	One of the language filter constants. Traditional Chinese.
FilterCJK	15	One of the language filter constants. Chinese, Japanese and Korean together.
FilterJapanese	4	One of the language filter constants. Japanese
FilterKorean	8	One of the language filter constants. Korean
FilterNonCJK	16	One of the language filter constants. Non Chineses/Japanes/Korean languages.

Chapter 9

Cocoa Controls

9.1 class DesktopTextArea

9.1.1 class DesktopTextArea

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in textarea class in Xojo.

Example:

```
// make a PDF from a textarea in Cocoa Xojo target:

// find view
Var n as NSViewMBS = TextArea1.NSViewMBS
if n = nil then
  MsgBox "Only in Cocoa!"
  Return
end if

// make pdf data
Var s as string = n.dataWithPDFInsideRect(0,0,n.frame.Width, n.frame.Height)

// save
Var f as FolderItem = GetSaveFolderItem("", "test.pdf")

if f<>Nil then

  Var b as BinaryStream = BinaryStream.Create(f, true)

  b.Write s

end if
```

Notes: Requires Xojo 2021r3 or newer.

9.1.2 Methods

9.1.3 SetTextThreadSafeMBS(text as string)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets text property on main thread.

Example:

```
// called in a thread:  
TextArea1.SetTextThreadSafeMBS "Hello World"
```

Notes: This method is to allow you to set the text property of a label in a thread without a problem.

If called on main thread, the plugin will simply set text property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

9.2 class TextArea

9.2.1 class TextArea

Plugin Version: 9.7, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in textarea class in Xojo.

Example:

```
// make a PDF from a textarea in Cocoa Xojo target:

// find view
Var n as NSViewMBS = TextArea1.NSViewMBS
if n = nil then
  MsgBox "Only in Cocoa!"
Return
end if

// make pdf data
Var s as string = n.dataWithPDFInsideRect(0,0,n.frame.Width, n.frame.Height)

// save
Var f as FolderItem = GetSaveFolderItem("", "test.pdf")

if f<>Nil then

  Var b as BinaryStream = BinaryStream.Create(f, true)

  b.Write s

end if
```

Notes: Requires RB 2009r4 or newer.

9.2.2 Methods

9.2.3 SetTextThreadSafeMBS(text as string)

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets text property on main thread.

Example:

```
// called in a thread:
TextArea1.SetTextThreadSafeMBS "Hello World"
```

Notes: This method is to allow you to set the text property of a label in a thread without a problem.

If called on main thread, the plugin will simply set text property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Chapter 10

Common Types

10.1 Globals

10.1.1 MakeDoublePointMBS(x as Double, y as Double) as DoublePointMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new double point.

Notes: Returns nil on low memory.

10.1.2 MakeDoubleRectMBS(left as Double, top as Double, width as Double, height as Double) as DoubleRectMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new double rectangle.

Notes: Returns nil on low memory.

10.1.3 MakeIntegerPointMBS(x as Integer, y as Integer) as IntegerPointMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new integer point.

Notes: Returns nil on low memory.

10.1.4 MakeIntegerRectMBS(left as Integer, top as Integer, width as Integer, height as Integer) as IntegerRectMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new integer rectangle.

Notes: Returns nil on low memory.

10.2 class DoublePointMBS

10.2.1 class DoublePointMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a double point.

10.2.2 Methods

10.2.3 Move(deltax as Double, deltay as Double)

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the point.

10.2.4 Properties

10.2.5 x as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The X property of the point.

Notes: (Read and Write property)

10.2.6 y as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Y property of the point.

Notes: (Read and Write property)

10.3 class DoubleRectMBS

10.3.1 class DoubleRectMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a double rectangle.

10.3.2 Methods

10.3.3 Intersection(other as DoubleRectMBS) as DoubleRectMBS

Plugin Version: 7.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The intersection area of two rectangles.

Notes: Returns nil if no intersection was found.

10.3.4 Intersects(other as DoubleRectMBS) as boolean

Plugin Version: 7.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether two rectangle intersect.

Notes: Returns true if yes and no on false.

10.3.5 Move(deltaX as Double, deltaY as Double)

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the rectangle.

10.3.6 Properties

10.3.7 Bottom as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The bottom property of the rectangle.

Notes: (Read and Write property)

10.3.8 height as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The height property of the rectangle.

Notes: Setting this property changes the right property.
(Read and Write property)

10.3.9 left as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The left property of the rectangle.

Notes: (Read and Write property)

10.3.10 right as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The right property of the rectangle.

Notes: (Read and Write property)

10.3.11 Size as Double

Plugin Version: 7.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size of this rectangle.

Notes: (Read only property)

10.3.12 top as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The top property of the rectangle.

Notes: (Read and Write property)

10.3.13 width as Double

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The width property of the rectangle.

Notes: Setting this property changes the bottom property.
(Read and Write property)

10.4 class IntegerPointMBS

10.4.1 class IntegerPointMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for an integer point.

10.4.2 Methods

10.4.3 Move(deltaX as Integer, deltaY as Integer)

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the point.

10.4.4 Properties

10.4.5 x as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The X property of the point.

Notes: (Read and Write property)

10.4.6 y as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Y property of the point.

Notes: (Read and Write property)

10.5 class IntegerRectMBS

10.5.1 class IntegerRectMBS

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for an integer rectangle.

10.5.2 Methods

10.5.3 Intersection(other as IntegerRectMBS) as IntegerRectMBS

Plugin Version: 7.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The intersection area of two rectangles.

Notes: Returns nil if no intersection was found.

10.5.4 Intersects(other as IntegerRectMBS) as boolean

Plugin Version: 7.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether two rectangle intersect.

Notes: Returns true if yes and no on false.

10.5.5 Move(deltaX as Integer, deltaY as Integer)

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the rectangle.

10.5.6 Properties

10.5.7 bottom as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The bottom property of the rectangle.

Notes: (Read and Write property)

10.5.8 height as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The height property of the rectangle.

Notes: Setting this property changes the right property.
(Read and Write property)

10.5.9 left as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The left property of the rectangle.

Notes: (Read and Write property)

10.5.10 right as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The right property of the rectangle.

Notes: (Read and Write property)

10.5.11 Size as Integer

Plugin Version: 7.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size of this rectangle.

Notes: (Read only property)

10.5.12 top as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The top property of the rectangle.

Notes: (Read and Write property)

10.5.13 width as Integer

Plugin Version: 2.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The width property of the rectangle.

Notes: Setting this property changes the right property.
(Read and Write property)

Chapter 11

Controls

11.1 Globals

11.1.1 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as DesktopWindow)

Plugin Version: 22.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call ShowModalWithin method on main thread.

Notes: This method is to allow you to call the ShowModalWithin method of a MessageDialog in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

Of course as this is asynchronously, you do not get details on which button was pressed.

Deprecated. Behavior from Xojo changed, so this method does no longer work.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

11.1.2 ShowModalThreadSafeMBS(extends theMessageDialog as MessageDialog)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call ShowModal method on main thread.

Notes: This method is to allow you to call the ShowModal method of a MessageDialog in a thread without

a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

Of course as this is asynchronously, you do not get details on which button was pressed.

Deprecated. Behavior from Xojo changed, so this method does no longer work.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0](#)
- [MBS Real Studio Plugins, version 13.0pr6](#)

Xojo Developer Magazine

- [11.2, page 10: News](#)

11.1.3 ShowModalWithinThreadSafeMBS(extends theMessageDialog as MessageDialog, parent as window)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call ShowModalWithin method on main thread.

Notes: This method is to allow you to call the ShowModalWithin method of a MessageDialog in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

Of course as this is asynchronously, you do not get details on which button was pressed.

Deprecated. Behavior from Xojo changed, so this method does no longer work.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [MBS Real Studio Plugins, version 13.0pr6](#)

11.2 class DesktopLabel

11.2.1 class DesktopLabel

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: An extension of Xojo's internal control.

11.2.2 Methods

11.2.3 SetTextThreadSafeMBS(text as string)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets text property on main thread.

Notes: This method is to allow you to set the text property of a label in a thread without a problem.

If called on main thread, the plugin will simply set text property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.3 class DesktopListbox

11.3.1 class DesktopListbox

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in listbox class in Xojo.

11.3.2 Methods

11.3.3 RefreshCellThreadSafeMBS(Row As Integer, Column As Integer)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Calls refresh cell on main thread.

Notes: This method is to allow you to invalidate a listbox cell in a thread without a problem.

If called on main thread, the plugin will simply call through directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.4 class DesktopProgressbar

11.4.1 class DesktopProgressbar

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in progressbar class in Xojo.

11.4.2 Methods

11.4.3 SetMaximumThreadSafeMBS(maximum as integer)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets maximum property on main thread.

Notes: This method is to allow you to set the maximum property of a progressbar in a thread without a problem.

If called on main thread, the plugin will simply set maximum property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.4.4 SetValueThreadSafeMBS(value as integer)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets value property on main thread.

Notes: This method is to allow you to set the value property of a progressbar in a thread without a problem.

If called on main thread, the plugin will simply set value property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.5 class DesktopTextField

11.5.1 class DesktopTextField

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in TextField class in Xojo.

11.5.2 Methods

11.5.3 SetTextThreadSafeMBS(text as string)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets text property on main thread.

Notes: This method is to allow you to set the text property of a label in a thread without a problem.

If called on main thread, the plugin will simply set text property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

11.6 class DesktopUIControl

11.6.1 class DesktopUIControl

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: An extension of Xojo's internal control.

11.6.2 Methods

11.6.3 RefreshThreadSafeMBS(immediately as Boolean = False)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call refresh method on main thread.

Notes: This method is to allow you to call the refresh method of a control in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

See also:

- 11.6.4 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately as Boolean = False) 257

11.6.4 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately as Boolean = False)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call refresh method on main thread.

Notes: This method is to allow you to call the refresh method of a control in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

See also:

- 11.6.3 RefreshThreadSafeMBS(immediately as Boolean = False) 257

11.6.5 SetEnabledThreadSafeMBS(value as boolean)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets enabled property on main thread.

Notes: This method is to allow you to set the enabled property of a control in a thread without a problem.

If called on main thread, the plugin will simply set enabled property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.6.6 SetVisibleThreadSafeMBS(value as boolean)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets visible property on main thread.

Notes: This method is to allow you to set the visible property of a control in a thread without a problem.

If called on main thread, the plugin will simply set visible property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.7 class Label

11.7.1 class Label

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: An extension of Xojo's internal control.

11.7.2 Methods

11.7.3 SetTextThreadSafeMBS(text as string)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets text property on main thread.

Notes: This method is to allow you to set the text property of a label in a thread without a problem.

If called on main thread, the plugin will simply set text property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

11.8 class Listbox

11.8.1 class Listbox

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in listbox class in Xojo.

11.8.2 Methods

11.8.3 InvalidateCellThreadSafeMBS(Row as Integer, Column as Integer)

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Calls invalide cell on main thread.

Notes: This method is to allow you to invalidate a listbox cell in a thread without a problem.

If called on main thread, the plugin will simply call through directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 13.5pr1](#)

11.9 class ProgressBar

11.9.1 class ProgressBar

Plugin Version: 9.7, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in progressbar class in Xojo.

11.9.2 Methods

11.9.3 SetMaximumThreadSafeMBS(maximum as Integer)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets maximum property on main thread.

Notes: This method is to allow you to set the maximum property of a progressbar in a thread without a problem.

If called on main thread, the plugin will simply set maximum property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [Thread GUI Access](#)
- [MBS Real Studio Plugins, version 13.0pr3](#)

11.9.4 SetMinimumThreadSafeMBS(minimum as Integer)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets minimum property on main thread.

Notes: This method is to allow you to set the minimum property of a progressbar in a thread without a problem.

If called on main thread, the plugin will simply set minimum property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [Thread GUI Access](#)
- [MBS Real Studio Plugins, version 13.0pr3](#)

11.9.5 SetValueThreadSafeMBS(value as Integer)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets value property on main thread.

Notes: This method is to allow you to set the value property of a progressbar in a thread without a problem.

If called on main thread, the plugin will simply set value property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [Thread GUI Access](#)
- [MBS Real Studio Plugins, version 13.0pr3](#)

11.10 class RectControl

11.10.1 class RectControl

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: An extension of Xojo's internal control.

11.10.2 Methods

11.10.3 InvalidateThreadSafeMBS(EraseBackground as boolean = true)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call invalidate method on main thread.

Notes: This method is to allow you to call the invalidate method of a control in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

See also:

- 11.10.4 InvalidateThreadSafeMBS(X as Integer, Y as Integer, Width as Integer, Height as Integer, EraseBackground as boolean = true) 263

11.10.4 InvalidateThreadSafeMBS(X as Integer, Y as Integer, Width as Integer, Height as Integer, EraseBackground as boolean = true)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call invalidate method on main thread.

Notes: This method is to allow you to call the invalidate method of a control in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

See also:

- 11.10.3 InvalidateThreadSafeMBS(EraseBackground as boolean = true) 263

11.10.5 RefreshThreadSafeMBS(EraseBackground as boolean = true)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Call refresh method on main thread.

Notes: This method is to allow you to call the refresh method of a control in a thread without a problem.

If called on main thread, the plugin will simply call method directly.

If called on other threads the plugin will schedule to call the method a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

11.10.6 SetEnabledThreadSafeMBS(value as boolean)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets enabled property on main thread.

Notes: This method is to allow you to set the enabled property of a control in a thread without a problem.

If called on main thread, the plugin will simply set enabled property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with CallDelegatesMBS.CallDelegateOnMainThreadMBS instead to do several update steps on one.

Blog Entries

- [MBS Real Studio Plugins, version 13.1pr9](#)
- [Thread GUI Access](#)
- [MBS Real Studio Plugins, version 13.0pr3](#)

11.10.7 SetVisibleThreadSafeMBS(value as boolean)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets visible property on main thread.

Notes: This method is to allow you to set the visible property of a control in a thread without a problem.

If called on main thread, the plugin will simply set visible property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [MBS Real Studio Plugins, version 13.1pr9](#)
- [MBS Real Studio Plugins, version 13.1pr1](#)
- [Thread GUI Access](#)
- [MBS Real Studio Plugins, version 13.0pr3](#)

11.11 class TextField

11.11.1 class TextField

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in TextField class in Xojo.

11.11.2 Methods

11.11.3 SetTextThreadSafeMBS(text as string)

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Sets text property on main thread.

Notes: This method is to allow you to set the text property of a label in a thread without a problem.

If called on main thread, the plugin will simply set text property directly.

If called on other threads the plugin will schedule to set the property a short time later on the main thread.

It may be better to call an update method you created on your window with `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead to do several update steps on one.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 13.2pr5](#)
- [Thread GUI Access](#)
- [MBS Real Studio Plugins, version 13.0pr3](#)

Chapter 12

CoreGraphics

Chapter 13

CPUInfo

13.1 class CPUIDMBS

13.1.1 class CPUIDMBS

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: A class around the CPUID command of x86 CPUs.

Example:

```
Var c as new CPUIDMBS
MsgBox c.BrandString
```

Notes: This class works only on x86 CPUs.

Values returned in the properties have very CPU vendor specific values.

So Intel and AMD use different meanings for a lot of values.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.3pr5](#)

Xojo Developer Magazine

- [5.4, page 50: The Case of the Top 20, The mayor is caught in an explosive situation, and our boys are on the case by Toby Rush](#)
- [5.3, page 6: News](#)
- [5.3, page 50: Poetry in Motion, Movin', movin', movin', get those icons movin'... by Toby Rush](#)

13.1.2 Methods

13.1.3 BrandString as String

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The brand string.

Notes: contains the brand string, e.g. "Intel(R) Xeon(TM) CPU 2.40GHz"

See also:

- 13.1.23 BrandString as String

274

13.1.4 CodeName as String

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The brief and human-friendly CPU codename, which was recognized.

13.1.5 CPUID(Selector as Integer) as boolean

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Calls CPUID for the given selector.

Example:

```
Var c as new CPUIDMBS
```

```
if c.CPUID(0) then
  MsgBox hex(c.EDX)
end if
```

Notes: Returns true on success and false on failure.

(always true on x86 CPUs and always false on PowerPC CPUs.)

Result values are stored in the four properties EAX, EBX, ECX and EDX.

13.1.6 ExtFamily as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: CPU extended family.

13.1.7 ExtModel as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: CPU extended model

13.1.8 Family as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: CPU family.

See also:

- 13.1.28 Family as Integer

275

13.1.9 FeatureName(index as Integer) as String

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the short textual representation of a CPU flag.

Notes: A constant string like "fpu", "tsc", "sse2", etc.

See kFeature* constants.

13.1.10 Flags(index as Integer) as Boolean

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Queries whether a given feature is available.

Example:

```
Var c as new CPUIDMBS
```

```

if c.Flags(CPUIDMBS.kFeatureLM) then
  MsgBox "64-bit CPU"
else
  MsgBox "32-bit CPU"
end if

```

Notes: See kFeature* constants.

13.1.11 L1DataCache as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: L1 data cache size in KB.

Notes: Could be zero, if the CPU lacks cache.
If the size cannot be determined, it will be -1.

13.1.12 L1InstructionCache as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: L1 instruction cache size in KB.

Notes: Could be zero, if the CPU lacks cache. If the size cannot be determined, it will be -1.
On some Intel CPUs, whose instruction cache is in fact a trace cache, the size will be expressed in K uOps.

13.1.13 L2Cache as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: L2 cache size in KB.

Notes: Could be zero, if the CPU lacks L2 cache.
If the size of the cache could not be determined, it will be -1

13.1.14 L3Cache as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: L3 cache size in KB.

Notes: Zero on most systems.

13.1.15 Model as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: CPU model.

See also:

- 13.1.29 Model as Integer

13.1.16 NumCores as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Number of CPU cores on the current processor.

13.1.17 NumLogicalCPUs as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Number of logical processors on the current processor.

Notes: Could be more than the number of physical cores, e.g. when the processor has HyperThreading.

13.1.18 Stepping as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: CPU stepping.

See also:

- 13.1.31 Stepping as Integer

277

13.1.19 TotalLogicalCPUs as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The total number of logical processors.

Notes: This is `num_logical_cpus * { total physical processors in the system }`

If you're writing a multithreaded program and you want to run it on all CPUs, this is the number of threads you need.

13.1.20 Vendor as Integer

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The recognized CPU vendor.

Notes: See `kVendor` constants.

13.1.21 VendorName as String

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The vendor name.

Notes: contains the CPU vendor string, e.g. "GenuineIntel"

13.1.22 Properties

13.1.23 BrandString as String

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: A string (47 characters maximum) with the brand name of the CPU.

Example:

```
Var c as new CPUIDMBS
MsgBox c.BrandString
```

Notes: Is "" if not supported.

(Read and Write property)

See also:

- 13.1.3 BrandString as String

270

13.1.24 EAX as UInt32

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The EAX register value after a CPUID function call.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.EAX)
```

Notes: (Read and Write property)

13.1.25 EBX as UInt32

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The EBX register value after a CPUID function call.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.EBX)
```

Notes: (Read and Write property)

13.1.26 ECX as UInt32

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The ECX register value after a CPUID function call.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.ECX)
```

Notes: (Read and Write property)

13.1.27 EDX as UInt32

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The EDX register value after a CPUID function call.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.EDX)
```

Notes: (Read and Write property)

13.1.28 Family as Integer

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The Family ID of the CPU.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.Family)
```

Notes: e.g. Family 6, Model 14 can be an Intel Core Duo CPU.
(Read and Write property)
See also:

- 13.1.8 Family as Integer

271

13.1.29 Model as Integer

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The Model ID of the CPU.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.Model)
```

Notes: Every CPU Family has several Models.
e.g. Family 6, Model 14 can be an Intel Core Duo CPU.
(Read and Write property)
See also:

- 13.1.15 Model as Integer

272

13.1.30 ProcessorVendor as String

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The Name of the Processor Vendor.

Example:

```
Var c as new CPUIDMBS
MsgBox c.ProcessorVendor
```

Notes: Possible values:

CPUID_VID_INTEL	"GenuineIntel"
CPUID_VID_UMC	"UMC UMC UMC "
CPUID_VID_AMD	"AuthenticAMD"
CPUID_VID_CYRIX	"CyrixInstead"
CPUID_VID_NEXGEN	"NexGenDriven"
CPUID_VID_CENTAUR	"CentaurHauls"
CPUID_VID_RISE	"RiseRiseRise"
CPUID_VID_SIS	"SiS SiS SiS "
CPUID_VID_TRANSMETA	"GenuineTMx86"
CPUID_VID_NSC	"Geode by NSC"

(Read and Write property)

13.1.31 Stepping as Integer

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The stepping ID of the CPU.

Example:

```
Var c as new CPUIDMBS
MsgBox str(c.Stepping)
```

Notes: Can be seen as a revision number of the processor.

(Read and Write property)

See also:

- 13.1.18 Stepping as Integer

13.1.32 Constants

Features

Constant	Value	Description
kFeature100MHzSteps	80	100 MHz multiplier control
kFeature3DNOW	56	AMD 3DNow! instructions supported
kFeature3DNOWEXT	57	AMD 3DNow! extended instructions supported
kFeature3DNOWPrefetch	68	PREFETCH/PREFETCHW support
kFeatureABM	65	LZCNT instruction support
kFeatureACPI	20	ACPI support (power states)
kFeatureAES	51	AES* instructions supported
kFeatureAPERFMPERF	91	MPERF/APERF MSRs support
kFeatureAPIC	9	APIC support
kFeatureAVX	54	Advanced vector extensions supported
kFeatureCID	40	Context ID supported
kFeatureCLFLUSH	18	CLFLUSH instruction supported
kFeatureCMOV	14	CMOVxx instructions supported
kFeatureCMP_LEGACY	63	core multi-processing legacy mode
kFeatureConstantTSC	82	TSC ticks at constant rate
kFeatureCPB	90	Core performance boost
kFeatureCX16	41	CMPXCHG16B instruction supported
kFeatureCX8	8	CMPXCHG8B instruction supported
kFeatureDCA	44	Direct cache access supported
kFeatureDE	2	Debugging extension
kFeatureDS_CPL	34	CPL Qualified Debug Store
kFeatureDTS	19	Debug store supported
kFeatureDTS64	32	64-bit Debug store supported
kFeatureEST	37	Enhanced SpeedStep
kFeatureF16C	87	16-bit FP convert instruction support
kFeatureFID	75	Frequency ID control
kFeatureFMA3	84	The FMA3 instruction set
kFeatureFMA4	85	The FMA4 instruction set
kFeatureFPU	0	Floating point unit
kFeatureFXSR	22	FXSAVE / FXRSTOR supported
kFeatureFXSR_OPT	59	FFXSR: FXSAVE and FXRSTOR optimizations
kFeatureHT	26	Hyper-threading supported (but might be disabled)
kFeatureHWPState	81	Hardware P-state control
kFeatureIA64	28	IA64 supported (Itanium only)
kFeatureIBS	70	Instruction-based sampling
kFeatureLAHF_LM	62	LAHF/SAHF supported in 64-bit mode
kFeatureLM	61	Long mode (x86_64/EM64T) supported
kFeatureMCA	13	Machine check architecture
kFeatureMCE	7	Machine check exception
kFeatureMisalignSSE	66	Misaligned SSE supported
kFeatureMMX	21	MMX instruction set supported
kFeatureMMXEXT	55	AMD MMX-extended instructions supported
kFeatureMONITOR	33	MONITOR / MWAIT supported
kFeatureMOVBE	49	MOVBE instruction supported
kFeatureMSR	5	Model-specific registers, RDMSR/WRMSR supported
kFeatureMTRR	10	Memory type range registers
kFeatureNX	58	No-execute bit supported
kFeatureOSVW	69	OS Visible Workaround (AMD)
kFeatureOSXSAVE	53	non-privileged copy of OSXSAVE supported
kFeaturePA	93	Processor accumulator
kFeaturePAE	6	Physical address extension
kFeaturePAT	15	Page attribute table
kFeaturePBE	29	Pending-break enable
kFeaturePCLMUL	31	PCLMULQDQ instruction supported
kFeaturePDCM	43	Performance capabilities MSR supported
kFeaturePFI	92	Processor Feedback Interface support
kFeaturePGE	12	Page global enable
kFeaturePN	17	Processor serial # implemented (Intel P3 only)

Vendors

Constant	Value	Description
kVendorAMD	1	AMD CPU
kVendorCentaur	6	x86 CPU by IDT
kVendorCyrix	2	Cyrix CPU
kVendorIntel	0	Intel CPU
kVendorNexGen	3	NexGen CPU
kVendorNSC	9	x86 CPU by National Semiconductor
kVendorRISE	7	x86 CPU by Rise Technology
kVendorSiS	8	x86 CPU by SiS
kVendorTransmeta	4	Transmeta CPU
kVendorUMC	5	x86 CPU by UMC
kVendorUnknown	-1	Unknown

Chapter 14

Currency, Date and Time Format

14.1 Globals

14.1.1 CDblMBS(text as string, byref value as Double, locale as string = "") as boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Parses a double value from a text with given locale.

Example:

```
Var value as Double
if CDblMBS("12,345", value, "de_DE") then
MsgBox str(value)
end if
```

Notes: Returns true on success and false on failure.

Value is set to the value detected.

Raises exception on invalid locale.

Blog Entries

- [MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0](#)
- [MBS Real Studio Plugins, version 13.0pr8](#)
- [CDbl and Format with locales](#)

Xojo Developer Magazine

- [11.2, page 10: News](#)

14.1.2 FormatDateMBS(format as string, value as date, locale as string = "") as string

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Formats a date with C time formatting functions.

Example:

```
// for Mac, Windows and Linux we usually have different locale names
Var fr as string
#if TargetMacOS then
fr = "fr_FR"
#elseif TargetWin32 then
fr = "fra"
#elseif TargetLinux
fr = "fr_FR.UTF8"
#else
?
#endif

Var d as new date
MsgBox FormatDateMBS("%x %X", d, fr)
```

Notes: Please use FormatDateTimeMBS function for iOS.

locale is the name of the locale to use. You can pass empty string to use default/current locale. Format is a format string like for strftime command in C.

The format specification is a string and may contain special character sequences called conversion specifications, each of which is introduced by a '%' character and terminated by some other character known as a conversion specifier character. All other character sequences are ordinary character sequences.

The characters of ordinary character sequences (including the null byte) are copied verbatim from format to s. However, the characters of conversion specifications are replaced as follows:

Some conversion specifications can be modified by preceding the conversion specifier character by the E or O modifier to indicate that an alternative format should be used. If the alternative format or specification does not exist for the current locale, the behavior will be as if the unmodified conversion specification were used. (SU) The Single UNIX Specification mentions %Ec, %EC, %Ex, %EX, %Ey, %EY, %Od, %Oe, %OH, %OI, %Om, %OM, %OS, %Ou, %OU, %OV, %Ow, %OW, %Oy, where the effect of the O modifier is to use alternative numeric symbols (say, roman numerals), and that of the E modifier is to use a locale-dependent alternative representation.

Blog Entries

- [MBS Xojo Plugins, version 19.5pr1](#)

- [MBS Xojo Plugins, version 17.3pr5](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.1](#)
- [MBS Xojo / Real Studio Plugins, version 14.2pr1](#)
- [Introducing FormatDateMBS function](#)
- [MBS Xojo / Real Studio Plugins, version 13.2pr1](#)

14.1.3 FormatDateTimeMBS(format as string, value as dateTime, locale as string = "") as string

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Formats a date with C time formatting functions.

Notes: Same as FormatDateMBS, but using DateTime class instead.

14.1.4 FormatMBS(format as string, value as Double, locale as string = "") as string

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Formats a double value.

Example:

```
MsgBox FormatMBS("%1.2f", 12.345, "de_DE")
```

Notes: locale is the name of the locale to use. You can pass empty string to use default/current locale. Format is a format string like for printf command in C.

The FormatMBS formats the value under control of the format. The format is a character string which contains three types of objects: plain characters, which are simply copied to standard output, character escape sequences which are converted and copied to the standard output, and format specifications, each of which causes printing of the next successive argument.

Character escape sequences are in backslash notation as defined in the ANSI X3.159-1989 ("ANSI C89"), with extensions. The characters and their meanings are as follows:

Each format specification is introduced by the percent character ("%"). The remainder of the format specification includes, in the following order:

Zero or more of the following flags:

Blog Entries

- [Introducing FormatDateMBS function](#)
- [MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0](#)
- [CDBl and Format with locales](#)
- [MBS Real Studio Plugins, version 13.0pr2](#)

Xojo Developer Magazine

- [11.2, page 10: News](#)

14.1.5 ParseDateMBS(format as string, text as string, byref value as date, locale as string = "") as boolean

Plugin Version: 17.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Parses a date.

Example:

```
Var s as string = "2013-11-12 18:31:01"
Var f as string = "%Y-%m-%d %H:%M:%S"
Var d as date
```

```
if ParseDateMBS(f, s, d) then
MsgBox d.LongDate + " " + d.LongTime
end if
```

Notes: Please use ParseDateTimeMBS function for iOS.

locale is the name of the locale to use. You can pass empty string to use default/current locale.
Format is a format string like for strftime command in C.

On success returns true. But even for half parsed dates you can find values in date.

Locale not supported on Windows.

The plugin does not support parsing time zones.

The ParseDateMBS function is the converse function to FormateDateMBS and converts the character string pointed to by s to values which are stored in the date, using the format specified by format. Here format is a character string that consists of field descriptors and text characters, reminiscent of scanf (in C++). Each field descriptor consists of a % character followed by another character that specifies the replacement for the field descriptor. All other characters in the format string must have a matching character in the input string, except for whitespace, which matches zero or more whitespace characters in the input string. There

should be whitespace or other alphanumeric characters between any two field descriptors.

The ParseDateMBS function processes the input string from left to right. Each of the three possible input elements (whitespace, literal, or format) are handled one after the other. If the input cannot be matched to the format string the function stops. The remainder of the format and input strings are not processed.

The supported input field descriptors are listed below. In case a text string (such as a weekday or month name) is to be matched, the comparison is case insensitive. In case a number is to be matched, leading zeros are permitted but not required.

Some field descriptors can be modified by the E or O modifier characters to indicate that an alternative format or specification should be used. If the alternative format or specification does not exist in the current locale, the unmodified field descriptor is used.

The E modifier specifies that the input string may contain alternative locale-dependent versions of the date and time representation:

Returns true on success. If the functions fails to match all of the format string and therefore an error occurred the function returns false.

Before libc 5.4.13 whitespace (and the 'n' and 't' specifications) was not handled, no 'E' and 'O' locale modifier characters were accepted, and the 'C' specification was a synonym for the 'c' specification.

The 'y' (year in century) specification is taken to specify a year in the 20th century by libc4 and libc5. It is taken to be a year in the range 1950-2049 by glibc 2.0. It is taken to be a year in 1969-2068 since glibc 2.1.

For reasons of symmetry, glibc tries to support for ParseDateMBS the same format characters as for FormatDateMBS. (In most cases the corresponding fields are parsed, but no field in tm is changed.) This leads to

Similarly, because of GNU extensions to FormatDateMBS, %k is accepted as a synonym for %H, and %l should be accepted as a synonym for %I, and %P is accepted as a synonym for %p. Finally

The glibc implementation does not require whitespace between two field descriptors.

Blog Entries

- [MBS Xojo Plugins, version 18.1pr5](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.1](#)
- [MBS Xojo Plugins, version 17.1pr3](#)

Xojo Developer Magazine

- [15.3, page 10: News](#)

14.1.6 ParseDateTimeMBS(format as string, text as string, byref value as DateTime, locale as string = "") as boolean

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Parses a date.

Notes: Same as ParseDateMBS, but using DateTime class instead.

14.2 class LocaleMBS

14.2.1 class LocaleMBS

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The plugin class which provides parameters for formatting numbers, especially currency values.

Example:

```
Var l as LocaleMBS = LocaleMBS.Locale("de_DE")
MsgBox l.CurrencySymbol
```

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0](#)
- [MBS Real Studio Plugins, version 13.0pr2](#)

Xojo Developer Magazine

- [11.2, page 10: News](#)

14.2.2 Methods

14.2.3 Constructor

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

14.2.4 Locale(Locale as string = "") as LocaleMBS

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries C locale for the given locale name.

Example:

```
Var l as LocaleMBS = LocaleMBS.Locale("de_DE")
MsgBox l.CurrencySymbol
```

Notes: Returns nil if no locale for that name was found.

If name is empty, you get current locale. If you use locale name "C", you get the default C locale. The locale name depends on the OS. For Mac and Linux you can see available locales in /usr/share/locale directory.

For Windows, please take a look here:

[http://msdn.microsoft.com/en-us/library/hzz3tw78\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/hzz3tw78(v=vs.71).aspx)

14.2.5 Properties

14.2.6 CurrencySymbol as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The local currency symbol.

Example:

```
Var l as LocaleMBS = LocaleMBS.Locale("de_DE")
MsgBox l.CurrencySymbol
```

Notes: (Read only property)

14.2.7 DecimalPoint as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The decimal point character, except for currency values, cannot be an empty string.

Notes: (Read only property)

14.2.8 FracDigits as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of digits after the decimal point in the local style for currency values.

Notes: (Read only property)

14.2.9 Grouping as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The sizes of the groups of digits, except for currency values.

Notes: This is a string where the asc value gives the actual value, representing group size from low order digit groups to high order (right to left). The list may be terminated with 0 or CHAR_MAX. If the list is terminated with 0, the last group size before the 0 is repeated to account for all the digits. If the list is terminated with CHAR_MAX, no more grouping is performed.

See example project for a function to decode.

(Read only property)

14.2.10 IntCurrSymbol as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The standardized international currency symbol.

Notes: (Read only property)

14.2.11 IntFracDigits as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of digits after the decimal point in an international-style currency value.

Notes: (Read only property)

14.2.12 IntNegCSPrecedes as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if the currency symbol precedes the currency value for negative values, false if it follows.

Notes: For internationally formatted monetary quantities.

(Read only property)

14.2.13 IntNegSepBySpace as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if a space is inserted between the currency symbol and the currency value for negative values, false otherwise.

Notes: For internationally formatted monetary quantities.
(Read only property)

14.2.14 IntNegSignPosition as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The location of the NegativeSign with respect to a negative quantity and the CurrencySymbol.

Notes: For internationally formatted monetary quantities.

Possible values:

(Read only property)

14.2.15 IntPosCSPrecedes as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if the currency symbol precedes the currency value for nonnegative values, false if it follows.

Notes: For internationally formatted monetary quantities.

(Read only property)

14.2.16 IntPosSepBySpace as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if a space is inserted between the currency symbol and the currency value for nonnegative values, false otherwise.

Notes: For internationally formatted monetary quantities.

(Read only property)

14.2.17 IntPosSignPosition as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The location of the PositiveSign with respect to a nonnegative quantity and the CurrencySymbol.

Notes: For internationally formatted monetary quantities.

Possible values:

(Read only property)

14.2.18 monDecimalPoint as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The decimal point character for currency values.

Notes: (Read only property)

14.2.19 monGrouping as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Like grouping but for currency values.

Notes: (Read only property)

14.2.20 monThousandsSep as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The separator for digit groups in currency values.

Notes: (Read only property)

14.2.21 Name as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the locale.

Notes: The name may be different to the one you asked for as some locales are mapped to others internally.
(Read only property)

14.2.22 NegativeSign as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The character used to denote negative currency values, usually a minus sign.

Notes: (Read only property)

14.2.23 NegCSPrecedes as Boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if the currency symbol precedes the currency value for negative values, false if it follows.

Notes: (Read only property)

14.2.24 NegSepBySpace as Boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if a space is inserted between the currency symbol and the currency value for negative values, false otherwise.

Notes: (Read only property)

14.2.25 NegSignPosition as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The location of the NegativeSign with respect to a negative quantity and the CurrencySymbol.

Notes: Possible values:

(Read only property)

14.2.26 PosCSPrecedes as Boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if the currency symbol precedes the currency value for nonnegative values, false if it follows.

Notes: (Read only property)

14.2.27 PositiveSign as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The character used to denote nonnegative currency values, usually the empty string.

Notes: (Read only property)

14.2.28 PosSepBySpace as Boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if a space is inserted between the currency symbol and the currency value for nonnegative values, false otherwise.

Notes: (Read only property)

14.2.29 PosSignPosition as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The location of the PositiveSign with respect to a nonnegative quantity and the CurrencySymbol.

Notes: Possible values:

(Read only property)

14.2.30 ThousandsSep as String

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The separator between groups of digits before the decimal point, except for currency values.

Notes: (Read only property)

- `%a` The abbreviated weekday name according to the current locale.
- `%A` The full weekday name according to the current locale.
- `%b` The abbreviated month name according to the current locale.
- `%B` The full month name according to the current locale.
- `%c` The preferred date and time representation for the current locale.
- `%C` The century number (year/100) as a 2-digit integer. (SU)
- `%d` The day of the month as a decimal number (range 01 to 31).
- `%D` Equivalent to `%m/%d/%y`. (Yecch-for Americans only. Americans should note that in other countries `%d/%m/%y` is rather common. This means that in international context this format is ambiguous and should not be used.) (SU)
- `%e` Like `%d`, the day of the month as a decimal number, but a leading zero is replaced by a space. (SU)
- `%E` Modifier: use alternative format, see below. (SU)
- `%F` Equivalent to `%Y-%m-%d` (the ISO 8601 date format). (C99)
- `%G` The ISO 8601 week-based year (see NOTES) with century as a decimal number. The 4-digit year corresponding to the ISO week number (see `%V`). This has the same format and value as `%Y`, except that if the ISO week number belongs to the previous or next year, that year is used instead. (TZ)
- `%g` Like `%G`, but without century, that is, with a 2-digit year (00-99). (TZ)
- `%h` Equivalent to `%b`. (SU)
- `%H` The hour as a decimal number using a 24-hour clock (range 00 to 23).
- `%I` The hour as a decimal number using a 12-hour clock (range 01 to 12).
- `%j` The day of the year as a decimal number (range 001 to 366).
- `%k` The hour (24-hour clock) as a decimal number (range 0 to 23); single digits are preceded by a blank. (See also `%H`.) (TZ)
- `%l` The hour (12-hour clock) as a decimal number (range 1 to 12); single digits are preceded by a blank. (See also `%I`.) (TZ)
- `%m` The month as a decimal number (range 01 to 12).
- `%M` The minute as a decimal number (range 00 to 59).
- `%n` A newline character. (SU)
- `%O` Modifier: use alternative format, see below. (SU)
- `%p` Either "AM" or "PM" according to the given time value, or the corresponding strings for the current locale. Noon is treated as "PM" and midnight as "AM".
- `%P` Like `%p` but in lowercase: "am" or "pm" or a corresponding string for the current locale. (GNU)
- `%r` The time in a.m. or p.m. notation. In the POSIX locale this is equivalent to `%I:%M:%S %p`. (SU)
- `%R` The time in 24-hour notation (`%H:%M`). (SU) For a version including the seconds, see `%T` below.
- `%s` The number of seconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC). (TZ)
- `%S` The second as a decimal number (range 00 to 60). (The range is up to 60 to allow for occasional leap seconds.)
- `%t` A tab character. (SU)
- `%T` The time in 24-hour notation (`%H:%M:%S`). (SU)
- `%u` The day of the week as a decimal, range 1 to 7, Monday being 1. See also `%w`. (SU)
- `%U` The week number of the current year as a decimal number, range 00 to 53, starting with the first Sunday as the first day of week 01. See also `%V` and

<code>\a</code>	Write a <bell>character.
<code>\b</code>	Write a <backspace>character.
<code>\c</code>	Ignore remaining characters in this string.
<code>\f</code>	Write a <form-feed>character.
<code>\n</code>	Write a <new-line>character.
<code>\r</code>	Write a <carriage return>character.
<code>\t</code>	Write a <tab>character.
<code>\v</code>	Write a <vertical tab>character.
<code>\'</code>	Write a <single quote>character.
<code>\\</code>	Write a backslash character.
<code>\num</code> or <code>\0num</code>	Write an 8-bit character whose ASCII value is the 1-, 2-, or 3-digit octal number num.

%%	The % character.
%a or %A	The weekday name according to the current locale, in abbreviated form or the full name.
%b or %B or %h	The month name according to the current locale, in abbreviated form or the full name.
%c	The date and time representation for the current locale.
%C	The century number (0-99).
%d or %e	The day of month (1-31).
%D	Equivalent to %m/%d/%y. (This is the American style date, very confusing to non-Americans, especially since %d/%m/%y is widely used in Europe. The ISO 8601 standard format is %Y-%m-%d.)
%H	The hour (0-23).
%I	The hour on a 12-hour clock (1-12).
%j	The day number in the year (1-366).
%m	The month number (1-12).
%M	The minute (0-59).
%n	Arbitrary whitespace.
%p	The locale's equivalent of AM or PM. (Note: there may be none.)
%r	The 12-hour clock time (using the locale's AM or PM). In the POSIX locale equivalent to %I:%M:%S %p. If t_fmt_ampm is empty in the LC_TIME part of the current locale then the behavior is undefined.
%R	Equivalent to %H:%M.
%S	The second (0-60; 60 may occur for leap seconds; earlier also 61 was allowed).
%t	Arbitrary whitespace.
%T	Equivalent to %H:%M:%S.
%U	The week number with Sunday the first day of the week (0-53). The first Sunday of January is the first day of week 1.
%w	The weekday number (0-6) with Sunday = 0.
%W	The week number with Monday the first day of the week (0-53). The first Monday of January is the first day of week 1.
%x	The date, using the locale's date format.
%X	The time, using the locale's time format.
%y	The year within century (0-99). When a century is not otherwise specified, values in the range 69-99 refer to years in the twentieth century (1969-1999); values in the range 00-68 refer to years in the twenty-first century (2000-2068).
%Y	The year, including century (for example, 1991).

%Ec	The locale's alternative date and time representation.
%EC	The name of the base year (period) in the locale's alternative representation.
%Ex	The locale's alternative date representation.
%EX	The locale's alternative time representation.
%Ey	The offset from %EC (year only) in the locale's alternative representation.
%EY	The full alternative year representation. The O modifier specifies that the numerical input may be in an alternative locale-dependent format:
%Od or %Oe	The day of the month using the locale's alternative numeric symbols; leading zeros are permitted but not required.
%OH	The hour (24-hour clock) using the locale's alternative numeric symbols.
%OI	The hour (12-hour clock) using the locale's alternative numeric symbols.
%Om	The month using the locale's alternative numeric symbols.
%OM	The minutes using the locale's alternative numeric symbols.
%OS	The seconds using the locale's alternative numeric symbols.
%OU	The week number of the year (Sunday as the first day of the week) using the locale's alternative numeric symbols.
%Ow	The number of the weekday (Sunday=0) using the locale's alternative numeric symbols.
%OW	The week number of the year (Monday as the first day of the week) using the locale's alternative numeric symbols.
%Oy	The year (offset from %C) using the locale's alternative numeric symbols.
%F	Equivalent to %Y-%m-%d, the ISO 8601 date format.
%g	The year corresponding to the ISO week number, but without the century (0-99).
%G	The year corresponding to the ISO week number. (For example, 1991.)
%u	The day of the week as a decimal number (1-7, where Monday = 1).
%V	The ISO 8601:1988 week number as a decimal number (1-53). If the week (starting on Monday) containing 1 January has four or more days in the new year, then it is considered week 1. Otherwise, it is the last week of the previous year, and the next week is week 1.
%z	An RFC-822/ISO 8601 standard timezone specification.
%Z	The timezone name.
%s	The number of seconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC). Leap seconds are not counted unless leap second support is available.

- 0 Parentheses around the entire string.
- 1 Before the string.
- 2 After the string.
- 3 Just before CurrencySymbol.
- 4 Just after CurrencySymbol.

- 0 Parentheses around the entire string.
- 1 Before the string.
- 2 After the string.
- 3 Just before CurrencySymbol.
- 4 Just after CurrencySymbol.

- 0 Parentheses around the entire string.
- 1 Before the string.
- 2 After the string.
- 3 Just before CurrencySymbol.
- 4 Just after CurrencySymbol.

- 0 Parentheses around the entire string.
- 1 Before the string.
- 2 After the string.
- 3 Just before CurrencySymbol.
- 4 Just after CurrencySymbol.

Chapter 15

Declare

15.1 class BlockMBS

15.1.1 class BlockMBS

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: A class to use blocks on Mac OS X with declares.

Blog Entries

- [Three new controls for iOS in Xojo](#)
- [MBS Xojo / Real Studio plug-ins version 16.2](#)
- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr4](#)

Xojo Developer Magazine

- [14.4, page 10: News](#)

15.1.2 Methods

15.1.3 Close

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Releases all blocks.

15.1.4 GetBlockB(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockB event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread. With running the event later, we can of course not return the event result to the block caller, but only the value in the AsyncBoolResult property.

15.1.5 GetBlockBI(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockBI event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread. With running the event later, we can of course not return the event result to the block caller, but only the value in the AsyncBoolResult property.

15.1.6 GetBlockBII(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockBII event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread. With running the event later, we can of course not return the event result to the block caller, but only the value in the AsyncBoolResult property.

15.1.7 GetBlockBIII(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockBIII event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread. With running the event later, we can of course not return the event result to the block caller, but only the value

in the AsyncBoolResult property.

15.1.8 GetBlockBIII(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockBIII event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread. With running the event later, we can of course not return the event result to the block caller, but only the value in the AsyncBoolResult property.

15.1.9 GetBlockV(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockV event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread.

15.1.10 GetBlockVI(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockVI event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread.

15.1.11 GetBlockVII(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockVII event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread.

15.1.12 GetBlockVIII(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockVIII event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread.

15.1.13 GetBlockVIII(tag as Variant = nil) as Integer

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Returns the address of a block to pass to a declare.

Notes: Tag is passed to the event.

Later when the block is invoked, the BlockVIII event is called. If the block is invoked on the main thread, we call the event directly. Else we schedule to call the event as soon as possible on the main thread.

15.1.14 Properties

15.1.15 AsyncBoolResult as Boolean

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: The result for a boolean block called on a non main thread.

Notes: If you use one of the BlockB methods to get a block and it's called on another thread but the main thread, we return the value of this property instead of the actual event result.

(Read only property)

15.1.16 Synchronous as Boolean

Plugin Version: 16.2, Platform: macOS, Targets: All.

Function: Whether to call events synchronously.

Notes: If the block is called on another thread, the plugin calls the event on the main thread.

If Synchronous is true, we call the main thread synchronously, else asynchronously.

Default is asynchronously to avoid dead locks.

(Read and Write property)

15.1.17 Events

15.1.18 BlockB(Async__ as boolean, tag as Variant) as boolean

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.

Tag: The tag value passed on block creation.

If you return a boolean, we pass it to the caller if `async=false`. For `Async = true`, the plugin already passed back `AsyncBoolResult` for you before this event is called.

15.1.19 BlockBI(Async__ as boolean, tag as Variant, value as Integer) as boolean

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.

Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to `Ptr`, `Boolean` or whatever data type you expect.

If you return a boolean, we pass it to the caller if `async=false`. For `Async = true`, the plugin already passed back `AsyncBoolResult` for you before this event is called.

15.1.20 BlockBII(Async__ as boolean, tag as Variant, value1 as Integer, value2 as Integer) as boolean

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.

Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to `Ptr`, `Boolean` or whatever data type you expect.

If you return a boolean, we pass it to the caller if `async=false`. For `Async = true`, the plugin already passed back `AsyncBoolResult` for you before this event is called.

15.1.21 BlockBIII(Async__ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer) as boolean

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.
Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to Ptr, Boolean or whatever data type you expect.

If you return a boolean, we pass it to the caller if `async=false`. For `Async = true`, the plugin already passed back `AsyncBoolResult` for you before this event is called.

15.1.22 BlockBIIII(Async__ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer, value4 as Integer) as boolean

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.
Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to Ptr, Boolean or whatever data type you expect.

If you return a boolean, we pass it to the caller if `async=false`. For `Async = true`, the plugin already passed back `AsyncBoolResult` for you before this event is called.

15.1.23 BlockV(Async__ as boolean, tag as Variant)

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.
Tag: The tag value passed on block creation.

15.1.24 BlockVI(Async__ as boolean, tag as Variant, value as Integer)

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.
Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to Ptr, Boolean or whatever data type you expect.

15.1.25 BlockVII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer)

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.

Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to Ptr, Boolean or whatever data type you expect.

15.1.26 BlockVIII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer)

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.

Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to Ptr, Boolean or whatever data type you expect.

15.1.27 BlockVIII(Async_ as boolean, tag as Variant, value1 as Integer, value2 as Integer, value3 as Integer, value4 as Integer)

Plugin Version: 13.4, Platform: macOS, Targets: All.

Function: Called when block is invoked.

Notes: Async: False if block is invoked directly on main thread. Else true so invoked later on main thread.

Tag: The tag value passed on block creation.

Value properties give the actual values. You may need to cast to Ptr, Boolean or whatever data type you expect.

15.2 class SoftDeclareMBS

15.2.1 class SoftDeclareMBS

Platforms: macOS, Linux, Windows, Targets: All.

Function: A way to do soft linking to libraries.

Example:

```
// Load the Stuffit Bundle on Mac OS X:
Var f as FolderItem
Var s as SoftDeclareMBS

f=FrameworksFolderMBS(-32765).Child("StuffIt.framework")
s=new SoftDeclareMBS

if s.LoadFrameworkFile(f) then
  MsgBox "ok"
end if
```

Notes: Deprecated. Please move to newer DeclareLibraryMBS and DeclareFunctionMBS classes.

You can make a declare like this:

```
Declare Function SpeakString lib "SpeechLib" (SpeakString as pstring) as Integer
```

But what if the SpeechLib is not installed?

Simple, but your application won't even launch on this machine.

Now you can of course use this MBS Plugin using the SpeechMBS functions, which are all weak linked.

But you can also use this class to weak link to the SpeechMBS library like this:

```
Var b as boolean, m,p as memoryblock, c as SoftDeclareMBS
```

```
c=new SoftDeclareMBS
p=newmemoryBlock(256) // make the string for the first parameter
p.pstring(0)="Hello World!"

m=newmemoryBlock(4) // make the memoryblock for the parameters
m.long(0)=p.Address(0) // set the first parameter to the address of the string buffer

if c.loadlibrary("SpeechLib") then // Load librarys
  if c.loadfunction("SpeakString") then // Load function
    b=c.Call(1,m)
  end if
end if
```

This way your application will be loaded, you can call the function, but people who don't have the SpeechLib will also here it.

Blog Entries

- [LiteSync and Xojo](#)

Xojo Developer Magazine

- [5.1, page 45: Detecting Rosetta, Are we running under emulation? by Christian Schmitz](#)

15.2.2 Methods

15.2.3 CallFunction(param as string,data as memoryblock) as boolean

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function.

Example:

```

Var c as SoftDeclareMBS
Var m,p,b as memoryBlock
Var f as folderItem
Var path as string

f=ApplicationsFolderMBS(-32766) // get a folder...
path=f.NativePath

MsgBox path

b=newmemoryBlock(1024)
b.long(0)=0 // make empty C string

p=newmemoryBlock(lenb(path)+3)
p.cstring(0)=path

m=newmemoryBlock(12+10)
m.long(0)=p.AddressMBS(0)
m.long(4)=b.AddressMBS(0)
m.long(8)=1023

c=new SoftDeclareMBS
if c.LoadDLL("KERNEL32") then
if c.loadfunction("GetShortPathNameA") then
c.CallingMode=0
MsgBox "found function"
if c.CallFunction("iii",m) then

```

```

msgbox "Short path is: "+b.cstring(0)
else
msgbox "Failed to call function."
end if
else
msgbox "Loading of function "+c.FunctionName+" failed."
end if
else
msgbox "Loading of Kernel32 failed."
end if

```

Notes: The param string is a combination of the characters "i" for integer, "l" for 64bit integer, "f" for single (float) and "d" for double.
Use "i" for booleans, shorts and pointers.

the memoryblock must match exactly the parameters you specified.
Returns true on success.
See also:

- 15.2.4 CallFunction(paramcount as Integer,data as memoryblock) as boolean

308

15.2.4 CallFunction(paramcount as Integer,data as memoryblock) as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function.

Example:

```

Var c as SoftDeclareMBS
Var m,p,b as memoryBlock
Var f as folderItem
Var path as string

f=ApplicationsFolderMBS(-32766) // get a folder...
path=f.NativePath

MsgBox path

b=newmemoryBlock(1024)
b.long(0)=0 // make empty C string

p=newmemoryBlock(lenb(path)+3)
p.cstring(0)=path

m=newmemoryBlock(12+10)
m.long(0)=p.AddressMBS(0)

```

```

m.long(4)=b.AddressMBS(0)
m.long(8)=1023

c=new SoftDeclareMBS
if c.LoadDLL("KERNEL32") then
if c.loadfunction("GetShortPathNameA") then
c.CallingMode=0
MsgBox "found function"
if c.CallFunction(3,m) then
msgbox "Short path is: "+b.cstring(0)
else
msgbox "Failed to call function."
end if
else
msgbox "Loading of function "+c.FunctionName+" failed."
end if
else
msgbox "Loading of Kernel32 failed."
end if

```

Notes: If paramtercount is 0, the memoryblock is ignored.

The size of the memoryblock must be minimum 4*paramcount.

Each parameter is set using m.long(n*4) where n=0 is the first parameter.

A parameter may be any integer value or an address of a memoryblock. The address can be read using memoryblock.addressMBS which is part of the plugin. You can even use only one memoryblock for all 3 parameters in the example like this:

```

m=newmemoryBlock(2100)
m.cstring(1024)=path
m.long(0)=m.address(20)
m.long(4)=m.address(1024)
m.long(8)=1024

```

First 12 bytes for the parameter table, the next 1000 bytes for the result buffer and finally a thousand bytes for the input string.

Before RB 3.1 this function was named "Call", but RB5 requires that the word "Call" is no longer valid for a function name.

Softdeclare is limited to only 6 parameters for plugin version 3.2. Plugin version 3.3 extends this to 8 parameters.

See also:

- 15.2.3 CallFunction(param as string,data as memoryblock) as boolean

15.2.5 CallFunctionDouble(param as string,data as memoryblock) as boolean

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function which returns a double value.

Notes: The param string is a combination of the characters "i" for integer, "l" for 64bit integer, "f" for single (float) and "d" for double.

Use "i" for booleans, shorts and pointers.

the memoryblock must match exactly the parameters you specified.

Returns true on success.

See also:

- 15.2.6 CallFunctionDouble(paramcount as Integer,data as memoryblock) as boolean 310

15.2.6 CallFunctionDouble(paramcount as Integer,data as memoryblock) as boolean

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function which returns a double value.

Notes: Fills the ResultDouble property.

If paramtercount is 0, the memoryblock is ignored.

The size of the memoryblock must be minimum 4*paramcount.

Each parameter is set using m.long(n*4) where n=0 is the first parameter.

A parameter may be any integer value or an address of a memoryblock. The address can be read using memoryblock.addressMBS which is part of the plugin.

Only 10 parameters can currently be used.

See also:

- 15.2.5 CallFunctionDouble(param as string,data as memoryblock) as boolean 310

15.2.7 CallFunctionInteger64(param as string,data as memoryblock) as boolean

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function which returns an integer value.

Notes: The param string is a combination of the characters "i" for integer, "l" for 64bit integer, "f" for single (float) and "d" for double.

Use "i" for booleans, shorts and pointers.

the memoryblock must match exactly the parameters you specified.

Returns true on success.

See also:

- 15.2. CLASS SOFTDECLAREMBS 311
- 15.2.8 CallFunctionInteger64(paramcount as Integer,data as memoryblock) as boolean 311

15.2.8 CallFunctionInteger64(paramcount as Integer,data as memoryblock) as boolean

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function which returns an integer value.

Notes: Fills the ResultInt64 property.

If paramtercount is 0, the memoryblock is ignored.

The size of the memoryblock must be minimum 4*paramcount.

Each parameter is set using m.long(n*4) where n=0 is the first parameter.

A parameter may be any integer value or an address of a memoryblock. The address can be read using memoryblock.addressMBS which is part of the plugin.

Only 10 parameters can currently be used.

See also:

- 15.2.7 CallFunctionInteger64(param as string,data as memoryblock) as boolean 310

15.2.9 CallMethod(param as string,data as memoryblock) as boolean

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function which returns no value.

Notes: The param string is a combination of the characters "i" for integer, "l" for 64bit integer, "f" for single (float) and "d" for double.

Use "i" for booleans, shorts and pointers.

the memoryblock must match exactly the parameters you specified.

Returns true on success.

See also:

- 15.2.10 CallMethod(paramcount as Integer,data as memoryblock) as boolean 311

15.2.10 CallMethod(paramcount as Integer,data as memoryblock) as boolean

Plugin Version: 7.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a function which returns no value.

Notes: If paramtercount is 0, the memoryblock is ignored.

The size of the memoryblock must be minimum 4*paramcount.

Each parameter is set using m.long(n*4) where n=0 is the first parameter.

A parameter may be any integer value or an address of a memoryblock. The address can be read using

memoryblock.addressMBS which is part of the plugin.

Only 10 parameters can currently be used.

Returns true on success.

See also:

- 15.2.9 CallMethod(param as string,data as memoryblock) as boolean

311

15.2.11 CopyLibrary(byref target as SoftDeclareMBS)

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the library handle to another softdeclare object.

Notes: if target is nil, a new object is created.

The library handle in the target object is set to point to the same library as the original object.

15.2.12 FreeLibrary as boolean

Plugin Version: 5.4, Platform: Windows, Targets: All.

Function: Releases the library.

Notes: Only for Windows currently this function releases the handles and unloads the library. Windows internally has a reference counter for the library so memory is only released when the last reference is freed. Lasterror is set.

15.2.13 LoadConstant(constname as string) as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads a constant from inside the library.

Example:

```

Var c as SoftDeclareMBS
Var s as CFStringMBS
Var m as MemoryBlock
Var handle as Integer

// Test Mac OS X
// Load the constant kABFirstNameProperty from the Addressbook framework

c=new SoftDeclareMBS

if c.LoadFramework("Addressbook.framework") then
msgbox "Loaded "+c.libname+" to "+format(c.libhandle,"-0")+""
if c.LoadConstant("kABFirstNameProperty") then

```



```

msgbox "Loaded constant "+c.ConstantName+" to "+format(c.ConstantPointer,"-0")+""
m=NewMemoryBlockFromPtrMBS(c.ConstantPointer) // I hope it's not nil

handle=m.Long(0)
if handle<>0 then
s=new CFStringMBS
s.Handle=handle

msgbox "Got value: "+s.str
end if
else
msgbox "Loading of constant "+c.ConstantName+" failed."
end if
else
msgbox "Loading of Addressbook.framework failed."
end if

```

Notes: Lasterror is set.

15.2.14 LoadDLL(libname as string) as boolean

Plugin Version: 5.4, Platform: Windows, Targets: All.

Function: Loads a Windows DLL.

Notes: Lasterror is set.

libname can be name (e.g. "KERNEL32"), filename (e.g. "KERNEL32.DLL") or path (e.g. "C:\WINDOWS\KERNEL32.DLL").

15.2.15 LoadDLLfromMemory(data as string) as boolean

Plugin Version: 7.7, Platform: Windows, Targets: All.

Function: Loads a Windows DLL from a string.

Notes: Some libraries don't like to be loaded from a string.

But else you can pass any DLL file content to this function.

The string is locked so it stays in memory.

On success the handle property is not zero and the function returns true.

15.2.16 LoadDylib(path as string) as boolean

Plugin Version: 5.4, Platform: macOS, Targets: All.

Function: Loads a library in Mac OS X dylib format.

Notes: Lasterror and Liberror are set.

15.2.17 LoadFramework(frameworkfilename as string) as boolean

Plugin Version: 5.4, Platform: macOS, Targets: All.

Function: Loads a framework in Mac OS X.

Example:

```
// A user's question:
// I am trying to call this to Carbon.framework using the softdeclare function:

// UInt32 SwapQDTextFlags(UInt32 newFlags);

// flags are OR'd

// kQDUseDefaultTextRendering = 0
// kQDUseTrueTypeScalerGlyphs = (1 «0)
// kQDUseCGTextRendering = (1 «1)
// kQDUseCGTextMetrics = (1 «2)
// kQDDontChangeFlags = 0xFFFFFFFF

// The call is to make the system use Quartz rendering for QuickDraw text (like the text in my WASTE-
// Field). How should I call this?

// The solution code:

Var s as SoftDeclareMBS
Var m as MemoryBlock

const flags=-1 // = 0xFFFFFFFF

s=new SoftDeclareMBS
m=NewMemoryBlock(10)
m.Long(0)=flags

if s.LoadFramework("Carbon.Framework") then
if s.LoadFunction("SwapQDTextFlags") then
if s.CallFunction(1,m) then
MsgBox str(s.Result) // returns 7 for me (using Silk)
end if
end if
```

```
end if
```

```
// Without any error checking!
```

Notes: frameworkfilename is e.g. "Carbon.framework"
Lasterror is set.

15.2.18 LoadFrameworkFile(folderitem) as boolean

Plugin Version: 3.2, Platform: macOS, Targets: All.

Function: Loads on Mac OS X a framework from the given file specification.

Example:

```
Var f as FolderItem
Var s as SoftDeclareMBS

f=SpecialFolder.Desktop.Child("spellcheck.bundle")
s=new SoftDeclareMBS

if s.LoadFrameworkFile(f) then
MsgBox "OK"
end if
```

Notes: Returns true if successful.
Lasterror is set.

15.2.19 LoadFunction(funcname as string) as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads a function from inside the library.

Example:

```
Var s as SoftDeclareMBS
Var m as MemoryBlock

const flags=-1 // = 0xFFFFFFFF

s=new SoftDeclareMBS
m=NewMemoryBlock(10)
m.Long(0)=flags

if s.LoadLibrary("Carbon.Framework") then
```

```

if s.LoadFunction("SwapQDTextFlags") then
if s.CallFunction(1,m) then
MsgBox str(s.Result) // returns 7 for me (using Silk)
end if
end if
end if

```

Notes: A user's question:

I am trying to call this to Carbon.framework using the softdeclare function:

```
UInt32 SwapQDTextFlags(UInt32 newFlags);
```

flags are OR'd

```

kQDUseDefaultTextRendering = 0
kQDUseTrueTypeScalerGlyphs = (1 «0)
kQDUseCGTextRendering = (1 «1)
kQDUseCGTextMetrics = (1 «2)
kQDDontChangeFlags = 0xFFFFFFFF

```

The call is to make the system use Quartz rendering for QuickDraw text (like the text in my WASTEField). How should I call this?

The solution code is above without any error checking!

Lasterror is set.

15.2.20 LoadLibrary(libname as string) as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads a library.

Notes: On Mac OS X e.g. "Carbon.framework" or "System.framework".

On Mac OS Carbon inside Classic e.g. "CarbonLib".

On Windows e.g. "KERNEL32" or "USER32".

Lasterror is set.

15.2.21 ParametersSupported(param as string) as boolean

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tests whether plugin supports the given parameter string.

Notes: The param string is a combination of the characters "i" for integer, "l" for 64bit integer, "f" for single (float) and "d" for double.

Use "i" for booleans, shorts and pointers.

Any new parameter string can be added. Please send an email to support to get a new combination added.

15.2.22 Properties

15.2.23 CallingMode as Integer

Plugin Version: 7.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: What calling mode to use.

Notes: 0 = Pascal (default)

1 = C

The Windows API works with Pascal, but calls to QuickTime DLL use C.
(Read and Write property)

15.2.24 ConstantFound as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Was the constant loaded?

Notes: Set by the LoadConstant function.

(Read and Write property)

15.2.25 ConstantName as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the constant to load.

Notes: Set by the LoadConstant function.

(Read and Write property)

15.2.26 ConstantPointer as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: The pointer of the constant loaded.

Notes: Set by the LoadConstant function.
(Read and Write property)

15.2.27 FunctionFound as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Was the function loaded?

Notes: Set by the LoadFunction function.
(Read only property)

15.2.28 FunctionName as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the function to load.

Notes: Set by the LoadFunction function.
(Read and Write property)

15.2.29 FunctionPointer as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: The pointer of the function loaded.

Notes: Set by the LoadFunction function.
(Read and Write property)

15.2.30 Lasterror as Integer

Plugin Version: 5.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error code.

Notes: LoadLibraryFile, LoadLibrary, LoadFunction, LoadConstant and FreeLibrary set this property.
(Read and Write property)

15.2.31 Liberror as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: An error message from loadlibrary.

Notes: Set by the LoadLibrary function.

(Read and Write property)

15.2.32 Libfound as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Was the library loaded?

Notes: Set by the LoadLibrary function.

(Read only property)

15.2.33 Libhandle as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: The handle of the library loaded.

Notes: Set by the LoadLibrary function.

On Mac OS Classic a CFragConnectionID.

On Mac OS X a CFBundleRef.

On Windows a HINSTANCE.

(Read and Write property)

15.2.34 Libname as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the library to load.

Notes: Set by the LoadLibrary function.

(Read and Write property)

15.2.35 Result as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: The result of the call function.

Notes: Set by the Call function.

(Read and Write property)

15.2.36 ResultDouble as Double

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The result of the call function CallFunctionDouble.

Notes: (Read and Write property)

15.2.37 ResultInt64 as MemoryBlock

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The result of the call function CallFunctionInteger64.

Example:

```
Var s as SoftDeclareMBS
```

```
Var m as MemoryBlock
```

```
s=new SoftDeclareMBS
```

```
if s.LoadDLL("test64bit.dll") then
```

```
if s.LoadFunction("Get64bitNumber") then
```

```
if s.CallFunctionInteger64(0,nil) then
```

```
MsgBox hex(s.ResultInt64.long(4))+hex(s.ResultInt64.long(0))
```

```
end if
```

```
end if
```

```
end if
```

```
' DLL was created with a function like this:
```

```
' IMPEXP long long Get64bitNumber()
```

```
' {
```

```
' return 0x1122334455667788;
```

```
' }
```

Notes: (Read and Write property)

Chapter 16

Disassembler

16.1 Globals

16.1.1 DisAssembleMBS

Plugin Version: 3.4, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Disassembles all further instructions from the current method following this command.

Notes: The result is stored inside the plugin memory and you can get a copy using the GetDisAssembleMBS function.

Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

16.1.2 DisAssembleObjectMethodMBS(target as object, Declaration as string)

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Disassembles all instructions from the given method inside the given object.

Notes: The result is stored inside the plugin memory and you can get a copy using the GetDisAssembleMBS function.

Requires Xojo 5.5 or newer.

Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

16.1.3 GetDisAssembleMBS as string

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns a string with the disassembled text.

Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

Chapter 17

DynamicDeclares

17.1 class DeclareCallbackMBS

17.1.1 class DeclareCallbackMBS

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a callback for use in declares.

Notes: Allows you to dynamically create callback functions.

Pass the function pointer to a C function which later calls back your callback.

Please make sure this callback object stays alive, e.g. stored in a global property. When the object is released and the callback later invoked, the application will crash.

When you code sign on macOS, you may need to use the entitlement for just-in-time compiler and for unsigned executable memory, e.g. the com.apple.security.cs.allow-unsigned-executable-memory entitlement.

Blog Entries

- [MBS Xojo Plugins, version 21.2pr1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [MBS Xojo Plugins, version 21.1pr7](#)
- [News from the MBS Xojo Plugins Version 20.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.3](#)
- [Dynamic Declare for Xojo](#)
- [MBS Xojo Plugins, version 20.3pr1](#)

Xojo Developer Magazine

- [21.1, page 28: News from MBS Xojo Plugins, What's up with MonkeyBread Software by Stefanie Juchmes](#)
- [19.3, page 10: News](#)

17.1.2 Methods

17.1.3 Constructor(Signature as String)

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new callback object.

Notes: Please pass a valid signature, e.g. "(pi)v" for a function taking one pointer, one integer and return nothing (void).

If the signature doesn't match the values put on the stack, the app may crash.

17.1.4 ParameterType(Index as Integer) as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries type of a parameter.

Notes: Index in range from 0 to ParameterCount-1.

17.1.5 Properties

17.1.6 AllowAsync as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the callbacks are allowed to go asynchronously.

Notes: For MacOS/iOS only.

Asynchronous callbacks have no chance to return a value as the call to the Xojo event happens later and the C function returns before the Xojo event is started.

Synchronous callbacks would let the callback wait for the Xojo main thread to react to the event and then return the result back.

AllowAsync is set to true automatically for callbacks with void as return automatically.
(Read and Write property)

17.1.7 CallCount as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The call counter.

Notes: We call how often the callback is called.

(Read only property)

17.1.8 FunctionPtr as Ptr

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The function pointer for the callback.

Notes: Pass this to the C functions to pass a callback function pointer.

(Read only property)

17.1.9 Name as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the callback.

Notes: You can set this to later read value in debugging.

(Read and Write property)

17.1.10 ParameterCount as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of parameters.

Notes: We currently limit this to 32 parameters.

(Read only property)

17.1.11 Signature as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The signature used to initialize this object.

Notes: (Read only property)

17.1.12 SignatureParameters as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The signature part for the parameters.

Notes: (Read only property)

17.1.13 SignatureReturn as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The signature part for the return value.

Notes: (Read only property)

17.1.14 Tag as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag value.

Notes: Store whatever you need to keep referenced.

(Read and Write property)

17.1.15 ParameterCFRetain(Index as Integer) as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether a parameter needs to be retained via CFRetain() call.

Notes: If the callback is within Apple's frameworks passing a CoreFoundation object, you can set this property to true, so we call CFRetain method on it in the callback and later release when the Xojo event was called.

This allows to avoid crashes with CoreFoundation objects passed from one thread to another thread for multi threaded callbacks.

Parameter must be of type pointer.

(Read and Write computed property)

17.1.16 ParameterCopyString(Index as Integer) as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether a parameter needs to be copied as a zero terminated string.

Notes: If the callback receives a C string, you can set this property to true, so we copy it in the callback and later release when the Xojo event was called.

This allows to avoid crashes with strings passed from one thread to another thread for multi threaded callbacks.

Parameter must be of type pointer or string.

(Read and Write computed property)

17.1.17 ParameterNSRetain(Index as Integer) as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether a parameter needs to be retained via [NSObject retain] call.

Notes: If the callback is within Apple's frameworks passing a Objective-C object, you can set this property to true, so we call retain method on it in the callback and later release when the Xojo event was called.

This allows to avoid crashes with NSObject objects passed from one thread to another thread for multi threaded callbacks.

Parameter must be of type pointer.

(Read and Write computed property)

17.1.18 Events

17.1.19 Callback(Parameters() as Variant) as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The callback event.

Notes: Parameters are converted to variants and passed as array.

The result is converted to the required data type and returned.

17.1.20 Constants

Call Modes

Constant	Value	Description
kCallModeARM	"A"	ARM calling
kCallModeARMThumb	"a"	ARM calling in thumb mode
kCallModeCDecl	"c"	x86 specific CDecl
kCallModeDefault	":"	Default calling conv (platform native)
kCallModeEllipsisVarargs	""	C ellipsis function call (variable/unnamed arguments (after , $\hat{\dots}$))
kCallModeSysCall	"\$ "	SysCall mode
kCallModeThisCallGNU	"#"	x86 specific, GNU C++ this calls are cdecl, but keep specific sig char for clarity.

Calling Conventions

Constant	Value	Description
kCallModeEllipsis	"e"	C ellipsis function call
kCallModeFastCallGNU	"f"	C x86 Windows GCC fast call
kCallModeFastCallMS	"F"	C x86 Windows Microsoft fast call
kCallModePrefix	"_"	The character to indicate that next character in the signature is a calling convention prefix.
kCallModeStdCall	"s"	C x86 Windows standard call
kCallModeThisCallMS	"+"	C x86 Windows Microsoft this call

Types

Constant	Value	Description
kTypeBool	"B"	bool
kTypeChar	"c"	char, 8 bit signed byte
kTypeDouble	"d"	Double, 64-bit and 8 bytes big.
kTypeEndArg	")"	end of arguments symbol.
kTypeFloat	"f"	Single, 32-bit and 4 bytes big.
kTypeInt	"i"	32-bit signed integer.
kTypeInt64	"l"	64-bit signed integer.
kTypeLong	"j"	32/64-bit signed integer. 64-bit only for MacOS and Linux in 64-bit application. Always 32-bit on Windows.
kTypePtr	"p"	Pointer. Maybe a reference to an object or array.
kTypeShort	"s"	16-bit signed integer.
kTypeString	"Z"	C String with zero byte as terminator.
kTypeStruct	"T"	Structure
kTypeUnsignedChar	"C"	8 bit unsigned byte
kTypeUnsignedInt	"I"	32-bit unsigned integer.
kTypeUnsignedInt64	"L"	64-bit unsigned integer.
kTypeUnsignedLong	"J"	32/64-bit unsigned integer. 64-bit only for MacOS and Linux in 64-bit application. Always 32-bit on Windows.
kTypeUnsignedShort	"S"	16-bit unsigned integer.
kTypeVoid	"v"	Nothing to return.

17.2 class DeclareFunctionMBS

17.2.1 class DeclareFunctionMBS

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a C function call.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")

// zlibVersion
// ZEXTERN Const char * ZEXPORT zlibVersion Of((void));

Var p As ptr = d.Symbol("zlibVersion")
Var f As New DeclareFunctionMBS("()Z", p)

Var n As String = f.Invoke
MsgBox "zlibVersion: "+n
```

Notes: Compared to built-in declares in Xojo, those functions allow a much more dynamic calling of functions with various calling conventions.

We have

- more dynamic
- the possibility to define declares at runtime
- more calling conventions
- support for variable arguments in C (dots in argument list)
- thread safe callbacks with DeclareCallBackMBS class.

Blog Entries

- [MBS Xojo Plugins, version 24.6pr1](#)
- [MBS Xojo Plugins, version 24.1pr1](#)
- [News from the MBS Xojo Plugins Version 20.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.3](#)
- [Dynamic Declare for Xojo](#)
- [MBS Xojo Plugins, version 20.3pr1](#)

17.2.2 Methods

17.2.3 ClearParameters

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears all parameters.

Notes: Sets them all to zero/nil.

17.2.4 Constructor(Signature as String, FunctionPtr as Ptr)

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")

// zlibVersion
// ZEXTERN Const char * ZEXPORT zlibVersion Of((void));

Var p As ptr = d.Symbol("zlibVersion")
Var f As New DeclareFunctionMBS("{}Z", p)

Var n As String = f.Invoke
MsgBox "zlibVersion: "+n
```

Notes: Pass the signature and function pointer from a C function.

If signature is incorrect, the application will probably crash with stack corruption.

17.2.5 Invoke as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Invokes the function.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")

// zlibVersion
// ZEXTERN Const char * ZEXPORT zlibVersion Of((void));

Var p As ptr = d.Symbol("zlibVersion")
```

```
Var f As New DeclareFunctionMBS("(Z)", p)
```

```
Var n As String = f.Invoke
MsgBox "zlibVersion: "+n
```

See also:

- 17.2.6 Invoke(Parameters() as Variant) as Variant

331

17.2.6 Invoke(Parameters() as Variant) as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Invokes the function.

Notes: First calls SetParameters with the given array to set parameters. Then invokes the function.

See also:

- 17.2.5 Invoke as Variant

330

17.2.7 SetParameters(paramArray Parameters as Variant)

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set all parameters together with passing arguments to this function.

Example:

```
// We try to run sysctlbyname on macOS
```

```
Var d As New DeclareLibraryMBS("/usr/lib/libc.dylib")
```

```
// int sysctlbyname(const char *name, void *oldp, size_t *oldlenp, void *newp, size_t newlen);
```

```
Var p As ptr = d.Symbol("sysctlbyname")
```

```
'const char* ->CString for name ->Z
```

```
'void* ->pointer to data ->p
```

```
'size_t * ->pointer to 32/64 bit integer for length ->p
```

```
'void* ->pointer to data ->p
```

```
'size_t ->32/64bit unsigned int ->L in 64-bit
```

```
Var f As New DeclareFunctionMBS("(ZpppL)i", p)
```

```
Var mData As New MemoryBlock(8)
```

```
Var mLen As New MemoryBlock(8)
```

```
Var pData As ptr = mData
```

```

Var pLen as ptr = mLen

mLen.UInt64Value(0) = mData.Size

f.SetParameters "hw.ncpu", pData, pLen, Nil, 0
Var n As Integer = f.Invoke
// result is zero on result or -1 on failure

Var CPUCount As Integer = mData.UInt32Value(0)

If n = 0 Then
MsgBox "CPU count: "+Str(CPUCount)
Else
MsgBox "Failed."
End If

Break // see in debugger

```

Notes: We automatically convert values based on the parameter type. The class stores reference to the variant when setting to make sure strings and MemoryBlocks are not freed too early.

Please pass C Strings as Xojo strings, not CString data type.
See also:

- 17.2.8 SetParameters(Parameters() as Variant)

332

17.2.8 SetParameters(Parameters() as Variant)

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets all arguments by passing an array of values.

Notes: We automatically convert values based on the parameter type.

The class stores reference to the variant when setting to make sure strings and MemoryBlocks are not freed too early.

Please pass C Strings as Xojo strings, not CString data type.
See also:

- 17.2.7 SetParameters(paramArray Parameters as Variant)

331

17.2.9 Properties

17.2.10 CallCount as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The call counter.

Notes: We call how often the function is called.

(Read only property)

17.2.11 CallMode as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The calling convention mode to use.

Notes: Default is to use default mode for the given platform.

(Read and Write property)

17.2.12 FunctionPtr as Ptr

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The pointer for the C function.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")

// zlibVersion
// ZEXTERN Const char * ZEXPORT zlibVersion Of((void));

Var p As ptr = d.Symbol("zlibVersion")

// call via delegate
' define delgate like this:
' Public Function zlibVersionDelegate() As CString

Var z As New zlibVersionDelegate(p)
Var v As String = z.Invoke
MsgBox "zlibVersion: "+v
```

Notes: (Read and Write property)

17.2.13 Name as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the function.

Notes: (Read and Write property)

17.2.14 ParameterCount as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of parameters.

Notes: We currently limit this to 32 parameters.

(Read only property)

17.2.15 Parameters as Dictionary

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries all parameters.

Notes: Mainly to inspect parameters in debugger.

Keys are indexes and values are the parameter values.

(Read only property)

17.2.16 Signature as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The signature used to initialize this object.

Notes: (Read only property)

17.2.17 SignatureParameters as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The signature part for the parameters.

Notes: (Read only property)

17.2.18 SignatureReturn as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The signature part for the return value.

Notes: (Read only property)

17.2.19 StackSize as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The stack size to use.

Notes: Defaults to 4096 bytes.

(Read and Write property)

17.2.20 Tag as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag value.

Notes: Store whatever you need to keep referenced.

(Read and Write property)

17.2.21 ParameterBoolean(Index as Integer) as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter as boolean.

Notes: You can get and set value here.

(Read and Write computed property)

17.2.22 ParameterDouble(Index as Integer) as Double

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter value as double.

Notes: You can get and set value here.

(Read and Write computed property)

17.2.23 ParameterInteger(Index as Integer) as Int64

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter as integer value.

Notes: You can get and set value here.

(Read and Write computed property)

17.2.24 ParameterPointer(Index as Integer) as Ptr

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter as pointer value.

Notes: You can get and set value here.

(Read and Write computed property)

17.2.25 ParameterSingle(Index as Integer) as Single

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter as single value.

Notes: You can get and set value here.

(Read and Write computed property)

17.2.26 ParameterString(Index as Integer) as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter as string value.

Example:

// We try to run strstr on C Library on macOS to find string in string with strstr as example:

```
Var d As New DeclareLibraryMBS("/usr/lib/libc.dylib")
```

```
// char *strstr(Const char * __big, Const char * __little);
```

```
Var p As ptr = d.Symbol("strstr")
```

```
Var f As New DeclareFunctionMBS("(ZZ)Z", p)
```

```
f.ParameterString(0) = "Hello World!"
```

```
f.ParameterString(1) = "World"
```

```
Var n1 As String = f.Invoke
```



```
// gives back World! as that was found

f.ParameterString(0) = "Hello World!"
f.ParameterString(1) = "xxx"

Var n2 As String = f.Invoke
// gives empty string as not found

Break // see in debugger
```

Notes: To avoid crashes, we only read strings from parameters with type string or pointer. But you can assign a string value to all types, which may not make sense for numbers. You can get and set value here.
(Read and Write computed property)

17.2.27 ParameterValue(Index as Integer) as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set the value as variant.

Notes: We automatically convert values based on the parameter type.

The class stores reference to the variant when setting to make sure strings and MemoryBlocks are not freed too early.

You can get and set value here.

Please pass C Strings as Xojo strings, not CString data type.
(Read and Write computed property)

17.2.28 Constants

Call Modes

Constant	Value	Description
kCallModeARM	"A"	ARM calling
kCallModeARMThumb	"a"	ARM calling in thumb mode
kCallModeCArm64	22	C arm64 call (AArch64)
kCallModeCArmArmhf	30	C arm call (arm hardŌˆCoat - e.g. raspberry pi)
kCallModeCDecl	"c"	x86 specific CDecl
kCallModeCDefault	0	C default function call for current platform
kCallModeCDefaultThis	99	C default function call for current platform with this pointer.
kCallModeCELLipsis	100	C ellipsis function call (named arguments (before ,Äˆ...,Äˆ))
kCallModeCELLipsisVarargs	101	C ellipsis function call (variable/unnamed arguments (after ,Äˆ...,Äˆ))
kCallModeCX64Win64	7	C x64 Windows standard call
kCallModeCX86Cdecl	1	C x86 platforms standard call
kCallModeCX86Win32FastGnu	4	C x86 Windows GCC fast call
kCallModeCX86Win32FastMs	3	C x86 Windows Microsoft fast call
kCallModeCX86Win32Std	2	C x86 Windows standard call
kCallModeCX86Win32This	70	C ellipsis function call (variable/unnamed arguments (after ,Äˆ...,Äˆ))
kCallModeCX86Win32ThisGnu	1	C x86 Windows GCC this call
kCallModeCX86Win32ThisMs	5	C x86 Windows Microsoft this call
kCallModeDefault	":"	Default calling conv (platform native)
kCallModeEllipsisVarargs	":"	C ellipsis function call (variable/unnamed arguments (after ,Äˆ...,Äˆ))
kCallModeSysCall	"\$ "	SysCall mode
kCallModeSysDefault	200	C default syscall for current platform
kCallModeSysX64SyscallSysV	204	C syscall for x64 System V platforms
kCallModeSysX86Int80HBSD	202	C syscall for x86 BSD platforms
kCallModeSysX86Int80HLinux	201	C syscall for x86 Linux
kCallModeThisCallGNU	"#"	x86 specific, GNU C++ this calls are cdecl, but keep specific sig char for clarity.

Calling Conventions

Constant	Value	Description
kCallModeEllipsis	"e"	C ellipsis function call
kCallModeFastCallGNU	"f"	C x86 Windows GCC fast call
kCallModeFastCallMS	"F"	C x86 Windows Microsoft fast call
kCallModePrefix	"_ "	The character to indicate that next character in the signature is a calling convention prefix.
kCallModeStdCall	"s"	C x86 Windows standard call
kCallModeThisCallMS	"+"	C x86 Windows Microsoft this call

Types

Constant	Value	Description
kTypeBool	"B"	bool
kTypeChar	"c"	char, 8 bit signed byte
kTypeDouble	"d"	Double, 64-bit and 8 bytes big.
kTypeEndArg	")"	end of arguments symbol.
kTypeFloat	"f"	Single, 32-bit and 4 bytes big.
kTypeInt	"i"	32-bit signed integer.
kTypeInt64	"l"	64-bit signed integer.
kTypeLong	"j"	32/64-bit signed integer. 64-bit only for MacOS and Linux in 64-bit application. Always 32-bit on Windows.
kTypePtr	"p"	Pointer. Maybe a reference to an object or array.
kTypeShort	"s"	16-bit signed integer.
kTypeString	"Z"	C String with zero byte as terminator.
kTypeStruct	"T"	Structure
kTypeUnsignedChar	"C"	8 bit unsigned byte
kTypeUnsignedInt	"I"	32-bit unsigned integer.
kTypeUnsignedInt64	"L"	64-bit unsigned integer.
kTypeUnsignedLong	"J"	32/64-bit unsigned integer. 64-bit only for MacOS and Linux in 64-bit application. Always 32-bit on Windows.
kTypeUnsignedShort	"S"	16-bit unsigned integer.
kTypeVoid	"v"	Nothing to return.

17.3 class DeclareLibraryMBS

17.3.1 class DeclareLibraryMBS

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a C library loaded into Xojo.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")
```

```
Var lines() As String = d.SymbolNames
```

```
Break // look in list of functions
```

Notes: Allows you to load a DLL on Windows, dylib on MacOS and shared object (so) on Linux and inspect the available functions and load functions by name.

Blog Entries

- [MBS Xojo Plugins, version 24.6pr1](#)
- [MBS Xojo Plugins, version 24.0pr8](#)
- [News from the MBS Xojo Plugins Version 20.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.3](#)
- [Dynamic Declare for Xojo](#)
- [MBS Xojo Plugins, version 20.3pr1](#)

17.3.2 Methods

17.3.3 Constructor(LibFile as folderItem)

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads a library.

Notes: Pass path to DLL on Windows, dylib on MacOS or so file on Linux. Raises exception in case of problems.

See also:

- [17.3.4 Constructor\(LibPath as string\)](#)

340

17.3.4 Constructor(LibPath as string)

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads a library.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")

Var lines() As String = d.SymbolNames
Break // look in list of functions
```

Notes: Pass path to DLL on Windows, dylib on MacOS or so file on Linux.
Raises exception in case of problems.
See also:

- 17.3.3 Constructor(LibFile as folderItem)

340

17.3.5 Symbol(name as string) as ptr

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries function or data pointer for a symbol.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")

// zlibVersion
// ZEXTERN Const char * ZEXPORT zlibVersion Of((void));

Var p As ptr = d.Symbol("zlibVersion")
Var f As New DeclareFunctionMBS("()Z", p)

Var n As String = f.Invoke
MsgBox "zlibVersion: "+n
```

Notes: Returns nil if not found.

17.3.6 SymbolName(index as Integer) as string

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries name of a symbol.

Notes: Index is in range from 0 to SymbolCount-1.

See also:

- 17.3.7 SymbolName(p as Ptr) as string

342

17.3.7 SymbolName(p as Ptr) as string

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries name of a symbol from the ptr.

Notes: Looks up the symbol table to find the entry with the right pointer and returns the name.

Returns "" if not found.

See also:

- 17.3.6 SymbolName(index as Integer) as string

341

17.3.8 SymbolNames as string()

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries list of symbols of a library.

Example:

```
// change path if you like to try this on Windows or Linux
Var d As New DeclareLibraryMBS("/usr/lib/libz.1.dylib")
```

```
Var lines() As String = d.SymbolNames
Break // look in list of functions
```

Notes: This may show more symbols as defined in public C headers.

17.3.9 Properties

17.3.10 Handle as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal object handle.

Notes: (Read only property)

17.3.11 Name as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries name of the library.

Notes: (Read only property)

17.3.12 Path as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries path of the library.

Notes: May return the path used to load the library.
(Read only property)

17.3.13 SymbolCount as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of symbols in the library.

Notes: (Read only property)

Chapter 18

Encryption and Hash

18.1 Globals

18.1.1 CalculateCRC16MemoryMBS(data as MemoryBlock, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided memoryblock.

Notes: Start is the start value.

Polynominal is what is xored to the value in each round.

FinalXOR is an XOR we do on the end.

If ReflectInput is true, we swap bits in input.

If ReflectOutput is true, we swap bits in output (before final XOR).

Returns CRC value.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.4pr5](#)

18.1.2 CalculateCRC16StringMBS(data as string, Start as UInt16 = 65535, Polynomial as UInt16 = &h1021, FinalXOR as UInt16 = 0, ReflectInput as boolean = false, ReflectOutput as boolean = false) as UInt16

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided string.

Example:

```

Var c as string = DecodeHex("3E4400026201000D")
Var p as UInt16 = CalculateCRC16StringMBS(c, &hFFFF, &h1021, 0, true, true)
MsgBox hex(p)+" = 3E5A"

```

Notes: Start is the start value.

Polynomial is what is xored to the value in each round.

FinalXOR is an XOR we do on the end.

If ReflectInput is true, we swap bits in input.

If ReflectOutput is true, we swap bits in output (before final XOR).

Returns CRC value.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.4pr5](#)

18.1.3 CRC16MBS(data as string) as UInt16

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided string.

Example:

```

Var n1 as Integer = CRC16MBS("Hello World")
Var n2 as Integer = CRC16MBS("Hallo World")

```

```
MsgBox str(n1)+" "+str(n2)
```

Notes: Please note: Different CRC functions give different result. This is one specific 16 bit CRC function which may be or not be the one you need for your application.

Blog Entries

- [MBS Real Studio Plugins, version 12.1pr1](#)

18.1.4 CRC_32InMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 32bit Checksum about the provided memory address.

18.1.5 CRC_32InMemMBS(address as Ptr, length as Integer) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 32bit Checksum about the provided memory address.

18.1.6 CRC_32OfStrContMBS(s as String, prevCRC as UInt32) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 32bit Checksum about the provided block of data.

18.1.7 CRC_32OfStrMBS(s as String) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 32bit Checksum about the provided string.

Example:

```
Var n as Integer
n=CRC_32OfStrMBS("Hello World")
' n is now 1243066710
```

Notes: See the text "About-CRC" for details about this Checksum things.
This function is also available as part of the memoryblock class.

About Checksums:

These functions calculate CRCs over a range of bytes in a MemoryBlock or in a String.

There are three versions of CRC calculation: One for 16 Bit, one for 32 Bit and one for 16 to 64 Bit wide checksums. If you are free to choose, I suggest that you use the 32 Bit version because of its accuracy and performance.

Some background on using checksums/CRCs:

Checksums, such as CRCs, are used to quickly verify that a chunk of data has not been modified somehow without your control. To use it, you'd calculate the checksum (CRC) of your data, then store that checksum value (which only needs 2 to 8 bytes of storage) somewhere. Later, when retrieving your data, you calculate its checksum again and compare its value with the previously stored value. If it does not match, the data got somehow corrupted. If it matches, that it is quite likely, although not 100% sure, that the data is still in its original state.

There are 3 different CRC algorithms available:

- `CRC_CCITT...()` as Integer
- `CRC_32...()` as Integer
- `CRC_Dillon...(bitWidth as Integer, ...)` as String

The CCITT version calculates a rather classic 16 bit CRC. Unless you need that CRC for legacy data, I recommend not to use it, but rather use the `CRC_32` version.

`CRC_32` is the most common used algorithm for 32 bit wide CRCs. Be aware there are theoretically many other ways to calculate a CRC 32, however.

The Dillon algorithm is a smart routine to calculate CRCs in any width between 16 and 64 bit. This one is only useful if you find that a 32 bit CRC is not sufficient for your needs. If you can live with a plain 32 bit CRC, you should prefer the `CRC_32` routines, because they are faster than the Dillon code.

The result, since it can be up to 64 bit in size, cannot be returned in an Integer (they can only hold 32 bit). Instead, the result is returned as a 8 byte value, stored in a 8 byte long string. To get the value of that string, you can copy the string into a `MemoryBlock`, and then access the parts of the 8 byte long value there. The demo project shows how to accomplish this.

The CRC code was developed and published by Matthew Dillon. Here's his web page with more information about it:

<http://www.backplane.com/diablo/crc64.html>

18.1.8 `CRC_CCITTInMemContMBS(address as Ptr, length as Integer, prevCRC as UInt32)` as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided memory address.

18.1.9 `CRC_CCITTInMemMBS(address as Ptr, length as Integer)` as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided memory address.

18.1.10 CRC_CCITTOfStrContMBS(s as String, prevCRC as UInt32) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided string.

18.1.11 CRC_CCITTOfStrMBS(s as String) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided string.

Example:

```
Var n as Integer
n=CRC_CCITTOfStrMBS("Hello World")
' n is now 39210
```

18.1.12 CRC_DillonInMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as String

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16 to 64bit Checksum about the provided memory address.

18.1.13 CRC_DillonOfStrMBS(bitWidth as Integer, s as String) as String

Plugin Version: 3.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16 to 64bit Checksum about the provided string.

Example:

```
Var s as string
s=CRC_DillonOfStrMBS(64,"Hello World")
' s has now the 64bit checksum inside an 8 byte binary string
```

18.1.14 CRC_DillonUInt64InMemMBS(bitWidth as Integer, address as Ptr, length as Integer) as UInt64

Plugin Version: 6.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16 to 64bit Checksum about the provided memory address.

18.1.15 CRC_DillonUInt64OfStrMBS(bitWidth as Integer, s as String) as UInt64

Plugin Version: 6.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16 to 64bit Checksum about the provided string.

18.1.16 GetHash32MBS(s as string) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates some special Hash value.

Example:

```
MsgBox hex(GetHash32MBS("xskin-b013fafit_01-PELVIS-BODY.skn")) // should show 67A53A4D
```

Notes: The 255 char limit was in the original C function, but should be resolved for this plugin, so string longer than 255 chars will work.

The original C code looks like this:

```
static UInt32 GetHash32 (StringPtr inString)
{

    int          length = inString [ 0 ] ;
    UInt32      hash = length + 1;
    int          i;

    for (i = 1; i <= length; i++)
    {
        hash = __rlwinm(hash, 3, 0, 31) ^tolower(inString [ i ] );
    }

    return hash;
}
```

Originally this was added for Cherie Benoit, which describes it like this:

"GetHash32MBS" duplicates the hashing function of Westlake Interactive's "Namer" application which creates LFN-style short filenames for use with the Macintosh version of "The Sims."

18.1.17 ModBusCalculateRTUMessageCRCMBS(data as string) as UInt16

Plugin Version: 16.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the RTU Message CRC string.

Example:

```

Var m as new MemoryBlock(9)

m.UInt8Value(0) = 1 // start of package
m.UInt8Value(1) = 6 // command code
m.UInt8Value(2) = 0 // some data...
m.UInt8Value(3) = 0
m.UInt8Value(4) = 0
m.UInt8Value(5) = &h21
m.UInt8Value(6) = 0 // checksum comes here
m.UInt8Value(7) = 0
m.UInt8Value(8) = 4 // end of package

Var Data as string = m.StringValue(0,6)
Var CheckSum as Int16 = ModBusCalculateRTUMessageCRCMBS(data)

m.UInt16Value(6) = checksum

MsgBox EncodeHex(m)

```

18.1.18 ValidateUUIDMBS(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Validates the given UUID/GUID.

Example:

```

// wrong
Var w1 as string = ValidateUUIDMBS("hello") // wrong due to missing {
Var w2 as string = ValidateUUIDMBS("550e8400-z29b-11d4-a716-446655440000",1) // wrong with, with z
Var w3 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d40716-446655440000 } ") // wrong with missing minus
Var w4 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d4-a716-44665544000 } ") // wrong with being too short
Var w5 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d4-a716-446655440000 } dssdsd",1) // wrong as it has extra chars on end
Var w6 as string = ValidateUUIDMBS("550e8400-e29b-11d4-a716-446655440000",1,4) // wrong as not version 4
Var w7 as string = ValidateUUIDMBS("6a12a4d5-e9e6-4568-ffcc-34c70b24a668",1,4) // wrong as not version 4

```

```

// okay
Var o1 as string = ValidateUUIDMBS("550e8400-e29b-11d4-a716-446655440000",1)
Var o2 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d4-a716-446655440000 } ")
Var o3 as string = ValidateUUIDMBS("6a12a4d5-e9e6-4568-afcc-34c70b24a668", 1)
Var o4 as string = ValidateUUIDMBS("6a12a4d5-e9e6-4568-afcc-34c70b24a668", 3)
Var o5 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d4-a716-446655440000 } ", 2)
Var o6 as string = ValidateUUIDMBS(" { 550e8400e29b11d4a716446655440000 } ", 2)
Var o7 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d40716-446655440000 } ",4) // wrong with miss-
ing minus, but fixed
Var o8 as string = ValidateUUIDMBS("550e8400-e29b-11d40716-446655440000 } ",4) // wrong with missing
{ , but fixed
Var o9 as string = ValidateUUIDMBS("6a12a4d5-e9e6-4568-afcc-34c70b24a668", 4, 4) // is version 4
Var o10 as string = ValidateUUIDMBS("6a12a4d5-e9e6-1568-afcc-34c70b24a668", 4, 1) // is version 1
Var o11 as string = ValidateUUIDMBS(" { 550e8400-e29b-11d4-a716-446655440000 } dssdsd",1+4) // wrong
as it has extra chars on end, but fixed

break // check in debugger

```

Notes: If the UUID is valid, you get it back.
If the UUID is invalid, you get an empty string back.

Pass 1 in mode to not require braces around UUID. Pass 2 to ignore minus characters. Pass 3 to combine those two.

You can add 4 to have the GUID fixed a bit like adding braces and minus chars if missing.

The requiredVersion parameter can be 1 to 5 to indicate the required GUID version you want to have.

Blog Entries

- [MBS Real Studio Plugins, version 13.0pr5](#)
- [MBS Real Studio Plugins, version 12.3pr11](#)

18.1.19 CRC_DillonUInt64MBS(extends mem as memoryblock, bitWidth as Integer, offset as Integer, numBytes as Integer) as UInt64

Plugin Version: 6.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16 to 64bit Checksum about the provided memory address.

Notes: The 64bit integer version for RB 2006r4 and newer.

18.1.20 CalculateTOTPMBS(Key as String, timeNow as UInt64, timeStart as UInt64, timeStep as UInt64, digitCount as Integer, DecodeBase32 as Boolean = false) as UInt32

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Calculates a Time-based one-time password.

Notes: Key is the secret to use.

timeNow the current time in seconds, e.g. pass CurrentUnixTimeMBS.

timeStart the start time in seconds, e.g. zero.

timeStep the duration the code is valid in seconds, e.g. 30 seconds.

digitCount: the number of digits to generate, e.g. 6.

Uses SHA-1 as hash algorithm.

Added DecodeBase32 parameter in v22.5 to specify whether the password is base32 encoded and should be decoded.

The result is returned as number and may miss leading zeros, if you just convert it to text.

Blog Entries

- [MBS Xojo Plugins, version 25.2pr3](#)
- [MBS Xojo Plugins, version 23.4pr1](#)
- [MBS Xojo Plugins in version 22.3](#)
- [MBS Xojo Plugins, version 22.3pr1](#)

18.1.21 CurrentUnixTimeMBS as UInt64

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries unix time stamp.

Notes: Seconds since 1970.

Useful for CalculateTOTPMBS function.

Blog Entries

- [MBS Xojo Plugins, version 22.3pr1](#)

18.1.22 DecodeFromBase32MBS(data as string) as String

Plugin Version: 23.4, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Decode a base 32 string.

Example:

```

Var s As String = EncodeToBase32MBS("The quick brown fox jumps over the lazy dog.")

If s = "KRUGKIDROVUWG2ZAMJZG653OEBTG66BANJ2W24DTEBXXMZLSEB2GQZJANRQXU6JAM-
RXWOLQ=" Then
  // okay
  Var d As String = DecodeFromBase32MBS(s)

  MsgBox d
Else
  Break
End If

```

Notes: You may need to use DefineEncoding to specify what encoding the decoded text should have.

Blog Entries

- [News from the MBS Xojo Plugins Version 23.4](#)
- [MBS Xojo Plugins, version 23.4pr3](#)

18.1.23 EncodeToBase32MBS(data as string) as String

Plugin Version: 23.4, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Encode a base32 string.

Example:

```

Var s As String = EncodeToBase32MBS("The quick brown fox jumps over the lazy dog.")

If s = "KRUGKIDROVUWG2ZAMJZG653OEBTG66BANJ2W24DTEBXXMZLSEB2GQZJANRQXU6JAM-
RXWOLQ=" Then
  // okay
  Var d As String = DecodeFromBase32MBS(s)

  MsgBox d
Else
  Break
End If

```

Notes: Please check what text encoding you want to use. Especially what the software processing the base32 string expects.

Blog Entries

- [News from the MBS Xojo Plugins Version 23.4](#)
- [MBS Xojo Plugins, version 23.4pr3](#)

18.2 class UUIDMBS

18.2.1 class UUIDMBS

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class to create Universally Unique Identifier.

Example:

```
Var u as UUIDMBS
```

```
u=new UUIDMBS
```

```
MsgBox EncodingToHexMBS(u.ValueString)
```

Notes: This class creates a 128 bit UUID (or UID or GUID) which is random.

On Mac OS X: Uses CoreFoundation system functions from Mac OS X 10.3.

On Linux: Uses unix system functions (libuuid).

On Windows: Uses UUID functions from RPC functions (Rpcrt4.dll).

If you need to validate a GUID or UUID, please check the Validate function.

Blog Entries

- [MBS Xojo Plugins, version 23.5pr1](#)
- [MBS Xojo Plugins, version 19.6pr1](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr4](#)
- [MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0](#)
- [MBS Real Studio Plugins, version 13.0pr5](#)
- [MBS REALbasic Plugins, version 10.6pr6](#)

18.2.2 Methods

18.2.3 randomUUID as UUIDMBS

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a random v4 GUID.

Example:

```
for i as Integer = 1 to 20
```

```
Listbox1.AddRow UUIDMBS.randomUUID.ValueFormattedString
```

[next](#)

Notes: This is using random numbers.
It is unlikely, but possible to have duplicates.
The UUID function checks for that.

18.2.4 UUID as UUIDMBS

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new UUID with system functions.

Notes: On Mac OS X: Uses CoreFoundation system functions from Mac OS X 10.3.

On Linux: Uses unix system functions (libuuid).

On Windows: Uses UUID functions from RPC functions (Rpcrt4.dll).

18.2.5 Validate(UUID as string, mode as Integer = 0, requiredVersion as Integer = 0) as string

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Validates the given UUID/GUID.

Example:

```
// wrong
Var w1 as string = UUIDMBS.Validate("hello") // wrong due to missing {
Var w2 as string = UUIDMBS.Validate("550e8400-z29b-11d4-a716-446655440000",1) // wrong with, with z
Var w3 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d40716-446655440000 } ") // wrong with missing minus
Var w4 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d4-a716-44665544000 } ") // wrong with being too short
Var w5 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d4-a716-446655440000 } dssdsd",1) // wrong as it has extra chars on end
Var w6 as string = UUIDMBS.Validate("550e8400-e29b-11d4-a716-446655440000",1,4) // wrong as not version 4
Var w7 as string = UUIDMBS.Validate("6a12a4d5-e9e6-4568-ffcc-34c70b24a668",1,4) // wrong as not version 4

// okay
Var o1 as string = UUIDMBS.Validate("550e8400-e29b-11d4-a716-446655440000",1)
Var o2 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d4-a716-446655440000 } ")
Var o3 as string = UUIDMBS.Validate("6a12a4d5-e9e6-4568-afcc-34c70b24a668", 1)
Var o4 as string = UUIDMBS.Validate("6a12a4d5-e9e6-4568-afcc-34c70b24a668", 3)
Var o5 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d4-a716-446655440000 } ", 2)
Var o6 as string = UUIDMBS.Validate(" { 550e8400e29b11d4a716446655440000 } ", 2)
```

```

Var o7 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d40716-446655440000 } ",4) // wrong with
missing minus, but fixed
Var o8 as string = UUIDMBS.Validate("550e8400-e29b-11d40716-446655440000 } ",4) // wrong with miss-
ing { , but fixed
Var o9 as string = UUIDMBS.Validate("6a12a4d5-e9e6-4568-afcc-34c70b24a668", 4, 4) // is version 4
Var o10 as string = UUIDMBS.Validate("6a12a4d5-e9e6-1568-afcc-34c70b24a668", 4, 1) // is version 1
Var o11 as string = UUIDMBS.Validate(" { 550e8400-e29b-11d4-a716-446655440000 } dssdsd",1+4) //
wrong as it has extra chars on end, but fixed

break // check in debugger

```

Notes: If the UUID is valid, you get it back.
If the UUID is invalid, you get an empty string back.

Pass 1 in mode to not require braces around UUID. Pass 2 to ignore minus characters. Pass 3 to combine those two.

You can add 4 to have the GUID fixed a bit like adding braces and minus chars if missing.

The requiredVersion parameter can be 1 to 5 to indicate the required GUID version you want to have.

18.2.6 ValueFormattedString(WithBrackets as Boolean = true) as String

Plugin Version: 11.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The UUID as a 38 character formatted hex string.

Example:

```

Var u as new UUIDMBS
MsgBox u.ValueFormattedString

```

Notes: example output: { D3ED4292-FC09-11DF-8AC5-7C6D628C4C29 }

Added WithBrackets parameter in version 22.1, so you can turn curly brackets off.

18.2.7 ValueHexString as String

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The UUID as a 32 character long hex string.

Example:

```

Var u as new UUIDMBS
MsgBox u.ValueHexString

```

Notes: example output: DCACAF4EFC0911DF92E37C6D628C4C29

18.2.8 ValueMemory as Memoryblock

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The UUID as a sixteen byte big memoryblock.

18.2.9 ValueString as String

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The UUID as a sixteen byte big binary string.

Example:

```
Var u as new UUIDMBS
MsgBox u.ValueString
```

Notes: This string is not for user display as it contains unreadable characters.

18.2.10 Properties

18.2.11 Lasterror as Integer

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Last error code.

Notes: -1 is for failure, 0 for okay.

Windows or Mac OS error codes else.

(Read and Write property)

18.2.12 Valid as Boolean

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the constructor was able to create an UUID.

Notes: (Read and Write property)

Chapter 19

Endian

19.1 Globals

19.1.1 EndianS16_BtoLMBS(n as Int16) as Int16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.2 EndianS16_BtoNMBS(n as Int16) as Int16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.3 EndianS16_LtoBMBS(n as Int16) as Int16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.4 EndianS16_LtoNMBS(n as Int16) as Int16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.5 EndianS16_NtoBMBS(n as Int16) as Int16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.6 EndianS16_NtoLMBS(n as Int16) as Int16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.7 EndianS32_BtoLMBS(n as Int32) as Int32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.8 EndianS32_BtoNMBS(n as Int32) as Int32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.9 EndianS32_LtoBMBS(n as Int32) as Int32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.10 EndianS32_LtoNMBS(n as Int32) as Int32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.11 EndianS32_NtoBMBS(n as Int32) as Int32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.12 EndianS32_NtoLMBS(n as Int32) as Int32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.13 EndianSwap16MBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Swaps a 16 bit integer.

19.1.14 EndianSwap32MBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Swaps a 32 bit integer.

19.1.15 EndianU16_BtoLMBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.16 EndianU16_BtoNMBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.17 EndianU16_LtoBMBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.18 EndianU16_LtoNMBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.19 EndianU16_NtoBMBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.20 EndianU16_NtoLMBS(n as UInt16) as UInt16

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.21 EndianU32_BtoLMBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.22 EndianU32_BtoNMBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.23 EndianU32_LtoBMBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.24 EndianU32_LtoNMBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Int32) as Int32

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.25 EndianU32_NtoBMBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

19.1.26 EndianU32_NtoLMBS(n as UInt32) as UInt32

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Notes: e.g.:

EndianS32_BtoNMBS(n as Integer) as Integer

EndianU16_LtoBMBS(n as Integer) as Integer

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Chapter 20

Filemapping and Shared Memory

20.1 class FileMappingMBS

20.1.1 class FileMappingMBS

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for file mapping and shared memory access.

Notes: FilemappingMBS is a clever way to map the content of a file into memory without loading it. All applications are today loaded via mapping using the virtual memory manager. So your data is inside the file but some memory is used to cache it and you can access it as a memoryblock.

You can use this class in several ways:

1. file mapping read/write: Call the constructor with a file, open the file mapping and map memory in your process to read or write a file.
2. file mapping read/write with temporary files: You can create (multi gigabyte) temporary memory storage to store data too big for your application own address space.
3. a variant of 2 is to pass nil to the constructor on Windows to have the data stored in the swap files.
4. use Constructor without paramaters and call CreateSharedMemory to create a shared memory object.
5. use Constructor without paramaters and call OpenSharedMemory to access a shared memory object from another process.

For shared memory objects, be careful how you design it. Your shared memory should have a flag for editing, so one app does not edit while another app edits. Add a version value to check your application versions. Also add a value for the revision of the content so your apps can see modifications. Finally your app should handle the possibility that the application crashes while writing data. So the data can be in a bad state.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.0](#)

- [Xojo 2020r2.1 arrived](#)
- [MBS Xojo Plugins, version 20.6pr2](#)
- [New in the MBS Xojo Plugins Version 20.2](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.2](#)
- [MBS Xojo Plugins, version 18.3pr5](#)
- [MBS Real Studio Plugins, version 13.1pr15](#)
- [MBS Real Studio Plugins, version 11.3pr14](#)
- [MBS REALbasic Plugins, version 10.5pr4](#)

Xojo Developer Magazine

- [19.2, page 10: News](#)

20.1.2 Methods

20.1.3 CloseFile

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Closes the file you used for backstore.

Notes: Called by the destructor automatically.

20.1.4 CloseFileMapping

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Closes the file mapping.

Notes: Called by the destructor automatically.

You close all views, than you close the file mapping and finally the file.

20.1.5 Constructor

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The dummy constructor used if you create a shared memory object.

Notes: Use this constructor if you continue with `OpenSharedMemory` or `CreateSharedMemory`.

See also:

- [20.1.6 Constructor\(file as folderitem, write as boolean = false\)](#)

20.1.6 Constructor(file as folderitem, write as boolean = false)

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens or creates a file for file mapping.

Notes: On Windows you can pass nil for the file parameter in order to have a file mapping using the swap files for back storage.

If write is true, the file is opened/created for write access.

See also:

- 20.1.5 Constructor

370

20.1.7 CreateSharedMemory(name as string, Size as Int64) as boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a shared memory object with the given name.

Notes: Returns true on success and false on failure. The name must be unique on the whole PC.

On success, you can call MapView to access memory.

isWriteable is set to true as shared memory objects are always read and write.

Fails if the shared memory object does exist. In this case, call OpenSharedMemory.

If your application crashes, on the next run the object will still exist on Mac OS X and Linux, so you need to open or delete & create it.

The Size should be a multiply of the page size (4096).

Sets DeleteSharedMemory to true so the object is deleted by the destructor. Set it to false if you don't want that behavior.

20.1.8 DeleteSharedMemory(name as string) as boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Targets: All.

Function: Deletes the shared memory object.

Notes: Returns true on success and false on failure.

Called by the destructor if DeleteSharedMemory property is true.

See also:

- 20.1.20 DeleteSharedMemory as Boolean

374

20.1.9 EnlargeFile(Size as Int64)

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Resizes the file to the given size.

Notes: This is for file mappings used as temporary storage with a temporary file. You use the Constructor to create a temporary file. This file is enlarged to the size you need. Next you call OpenFileMapping and MapView. To cleanup, you close all views and the file mapping. Now before you close the file, you should call ShrinkFile. ShrinkFile reduces the file size to zero so the operation system doesn't start flushing the shared memory to the file. You can get this automatically if you set ShrinkFileOnClose to true.

20.1.10 GetSharedMemoryValue(name as string) as MemoryBlock

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries value of shared memory object.

Notes: Returns copy of the data.

Since data is stored in blocks of page size (usually 4K), the returned size has additional null bytes on the end to round size up.

If you store text there, you may look for the end of your data, e.g. for JSON do a trim(Chr(0)) to remove the null bytes.

20.1.11 HasSharedMemoryValue(name as string) as Boolean

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether shared memory object exists.

20.1.12 MapView(mem as MemoryBlock, offset as Int64, Size as Integer) as FileMapViewMBS

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Maps a portion of the file mapping or shared memory object into your application's address space.

Notes: Returns nil on any error.

Using invalide offset/size values can lead into a bad mapping.

Here you can pass your own memoryblock for back storage.

Make sure your memory is 4 or 64 KB aligned, depending on platform.

(Windows need 64 KB, Mac only 4 KB)

Changed Size parameter from Int32 to Integer in version 21.0.

Xojo may not handle ptr with offsets >2 GB as of version 2020r2. This is fixed in 2020r2.1.
See also:

- 20.1.13 MapView(offset as Int64, Size as Integer) as FileMapViewMBS 373

20.1.13 MapView(offset as Int64, Size as Integer) as FileMapViewMBS

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Maps a portion of the file mapping or shared memory object into your application's address space.

Notes: Returns nil on any error.

Using invalide offset/size values can lead into a bad mapping.

(Windows need 64 KB, Mac only 4 KB)

Changed Size parameter from Int32 to Integer in version 21.0.

Xojo may not handle ptr with offsets >2 GB as of version 2020r2. This is fixed in 2020r2.1.

See also:

- 20.1.12 MapView(mem as MemoryBlock, offset as Int64, Size as Integer) as FileMapViewMBS 372

20.1.14 OpenFileMapping(MaxSize as Int64 = 0) as boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens a file mapping.

Notes: If MaxSize is zero, the file size is used for creating the mapping.

The Size should be a multiply of the page size (4096).

Returns true on success.

Do not call after using OpenSharedMemory or CreateSharedMemory as they open the file mapping for you.

20.1.15 OpenSharedMemory(name as string) as boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens a shared memory object with the given name.

Notes: Returns true on success and false on failure.

On success, you can call MapView to access memory.

isWriteable is set to true as shared memory objects are always read and write.

Fails if the shared memory object does not exist. In this case, call `CreateSharedMemory`.

20.1.16 `SetSharedMemoryValue(name as string, data as MemoryBlock)` as Boolean

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets shared memory value.

Notes: Persistent object on Mac/Linux till computer restarts or shuts down.

On Windows object is destroyed when last application referring it quits.

Returns true on success.

Size of the data block will be rounded up to page size.

On Windows the shared memory objects can't be accessed from another login session. To create a truly global one, we would need to run your application as administrator and prefix the name with "Global\".

20.1.17 `ShrinkFile`

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Resizes the file to be empty.

Notes: Useful if you used `EnlargeFile` before.

20.1.18 `Properties`

20.1.19 `DeleteFileOnClose` as Boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to delete the file in the destructor.

Notes: Useful if a temporary file is used as backstore.

(Read and Write property)

20.1.20 `DeleteSharedMemory` as Boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Targets: All.

Function: Whether to delete the shared memory object on closing.

Notes: True by default after Creating a shared memory object.

On Windows the shared memory objects exist as long as someone uses them.
On Mac OS X they live until you kill them even if your application is not running.
(Read and Write property)

See also:

- 20.1.8 DeleteSharedMemory(name as string) as boolean

371

20.1.21 File as FolderItem

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The file reference used for the backstore.

Notes: (Read only property)

20.1.22 isWriteable as Boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether opened/created the file mapping with write permission.

Notes: Opening files for reading only in order to read inside gives the operation system some optimization possibilities.

(Read only property)

20.1.23 Lasterror as Integer

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error code.

Notes: On Windows typically a Windows error code.

On Linux and Mac OS X typically 0 for success and other values for errors.

(Read and Write property)

20.1.24 LasterrorString as String

Plugin Version: 10.5, Platform: Windows, Targets: All.

Function: The string for the last error code.

Notes: Only implemented for Windows.

Returns "" on any error.

(Read only property)

20.1.25 Name as String

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name used for a named shared memory object.

Notes: (Read only property)

20.1.26 SharedMemorySize as Int64

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries shared memory object size.

Notes: Only valid for shared memory objects in handle.

(Read only property)

20.1.27 ShrinkFileOnClose as Boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to call ShrinkFile automatically from the destructor.

Notes: Default false.

(Read and Write property)

20.2 class FileMappingViewMBS

20.2.1 class FileMappingViewMBS

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a file mapping view.

Notes: You can use file mapping with a 10 GB bit file. Your application has only 4 GB of address space and effectively you can only use 2 GB. So you can try to map in a few hundred mega bytes at a given time. But you can move this view on the large file to read the whole file. But remember: File mapping is only efficient if you read a little data from the file and jump a lot. For reading the whole file, use a `binarystream`.

Blog Entries

- [MBS REALbasic Plugins, version 10.5pr4](#)

20.2.2 Methods

20.2.3 FlushView

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Flushes all data to the backstore.

Notes: For file mappings, the modified pages are written back to the file now. Normally you won't call this and let the operation system decide when to write the data to the file.

20.2.4 UnmapView

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Frees the memory allocated by the view.

Notes: Called automatically by the destructor.

20.2.5 Properties

20.2.6 FlushOnClose as Boolean

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to call `FlushView` for you in the destructor.

Notes: (Read and Write property)

20.2.7 Memory as Memoryblock

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The memoryblock for this view.

Example:

```
Var v as FileMappingViewMBS // your view
```

```
v.memory.cstring(0)="Hello World"
```

Notes: The memoryblock has no known size and it becomes invalid once this view object is destroyed. So keep a reference around as long as you use the memoryblock.

(Read only property)

20.2.8 Offset as Int64

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The offset used to create this view.

Notes: (Read only property)

20.2.9 Parent as FileMappingMBS

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The owner FileMapping object.

Notes: (Read only property)

20.2.10 Size as Integer

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size in bytes of this view.

Notes: (Read only property)

Chapter 21

Files

21.1 class DirectorySizeMBS

21.1.1 class DirectorySizeMBS

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for size information about a folder.

Example:

```
// chose a folder
Var f as FolderItem = SelectFolder

// calculate
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)

// display
MsgBox str(d.FilesCount)+" files in "+str(d.FolderCount)+" folder"
```

Blog Entries

- [MBS Xojo Plugins in version 22.4](#)
- [MBS Xojo Plugins, version 22.4pr1](#)
- [MBS Plugins updated for Xojo 2019r2](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4](#)
- [MBS Xojo / Real Studio Plugins, version 16.4pr4](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr9](#)
- [MBS Real Studio Plugins, version 13.1pr13](#)

- [MBS Real Studio Plugins, version 13.1pr3](#)
- [MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0](#)
- [MBS Plugins 11.1 Release notes](#)

Videos

- [Presentation from London conference about MBS Plugins.](#)

21.1.2 Methods

21.1.3 Add(d as DirectorySizeMBS)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds the values of the given directory size object to the current one.

Notes: Passing nil is okay and will be ignored.

This method was added to support counting several folders and adding the results to one central object.

21.1.4 close

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

Notes: There is no need to call this method except you want to free all resources of this object now without waiting for Xojo to do it for you.

21.1.5 Constructor

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

21.1.6 Update(folder as folderitem, recursive as boolean, ticks as Integer) as boolean

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Updates the class.

Example:

```

// chose a folder
Var f as FolderItem = SpecialFolder.Applications

// calculate
Var d as new DirectorySizeMBS

// update object
call d.Update(f, true, 0)

// display
MsgBox str(d.FilesCount)+" files and "+str(d.FolderCount)+" folders"

```

Notes: The folder specified is searched for files and folders. Normally you'd better use CalculateDirectorySizeMBS.

Ticks is the count of ticks (1/60th second) which must pass till time is given to other threads. (e.g. 10)
Returns true if successful.

If you call this function in a thread you can set the cancel property in a pushbutton event handler to stop this function.

21.1.7 Properties

21.1.8 Cancel as Boolean

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the Update function should stop as soon as possible.

Notes: (Read and Write property)

21.1.9 CompressedSize as UInt64

Plugin Version: 11.3, Platform: Windows, Targets: All.

Function: The size in bytes of all files.

Notes: If a file is compressed, we add here the compressed size.

If a file is not compressed, we add the normal logical size.

Only used on Windows and only with QueryCompressedSizes = true.

(Read only property)

21.1.10 CountBundlesAsItem as Boolean

Plugin Version: 11.1, Platform: macOS, Targets: All.

Function: Whether to count bundles as files.

Example:

```
// chose a folder
Var f as FolderItem = SpecialFolder.Applications

// calculate
Var d1 as new DirectorySizeMBS
d1.CountBundlesAsItem = true

call d1.Update(f, true, 0)

Var d2 as new DirectorySizeMBS
d2.CountBundlesAsItem = false

call d2.Update(f, true, 0)

// display
MsgBox "Normal count: "+str(D2.FilesCount)+" files and "+str(d2.FolderCount)+" folders"+EndOfLine+_
"With bundles as files: "+str(D1.FilesCount)+" files and "+str(d1.FolderCount)+" folders"
```

Notes: Bundles like applications are counted as a single file if this property is true or as a folder with files inside if this property is false.

Default is false.

(Read and Write property)

21.1.11 Directory as FolderItem

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The folder which was searched.

Notes: (Read and Write property)

21.1.12 FilesCount as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of files counted.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
```

```
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.FilesCount)+" files"
```

Notes: FilesCount=VisibleFilesCount+HiddenFilesCount
(Read only property)

21.1.13 FolderCount as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of folders counted.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.FolderCount)+" folder"
```

Notes: FolderCount=VisibleFolderCount+HiddenFolderCount
(Read only property)

21.1.14 HiddenCompressedSize as UInt64

Plugin Version: 11.3, Platform: Windows, Targets: All.

Function: The size in bytes of all hidden files.

Notes: If a file is compressed, we add here the compressed size.

If a file is not compressed, we add the normal logical size.

Only used on Windows and only with QueryCompressedSizes = true.

(Read only property)

21.1.15 HiddenFilesCount as Integer

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of hidden files counted.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.HiddenFilesCount)+" hidden files"
```

Notes: A file is invisible if the Invisible flag is set for this file.

(Read and Write property)

21.1.16 HiddenFolderCount as Integer

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of folders.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.HiddenFolderCount)+" hidden folders"
```

Notes: A file is invisible if the Invisible flag is set for this file.

(Read and Write property)

21.1.17 HiddenItemCount as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of invisible items.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.HiddenItemCount)+" hidden items"
```

Notes: An item is invisible if the Invisible flag is set for this file.

HiddenItemCount=HiddenFolderCount+HiddenFilesCount

(Read only property)

21.1.18 HiddenLogicalDataForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical size of the data forks of all hidden file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical data fork size of hidden files: "+Format(d.HiddenLogicalDataForkSize/1000000,"0")+
" MB"
```


Notes: (Read and Write property)

21.1.19 HiddenLogicalResourceForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical size of the resource forks of all hidden file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical resource fork size of hidden files: "+Format(d.HiddenLogicalResourceForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.20 HiddenLogicalTotalSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical size of the all forks of all hidden file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical total size of hidden files: "+Format(d.HiddenLogicalTotalSize/1000000,"0")+
" MB"
```

Notes: (Read only property)

21.1.21 HiddenPhysicalDataForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical size of the data forks of all hidden file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical data fork size of hidden files: "+Format(d.HiddenPhysicalDataForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.22 HiddenPhysicalResourceForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical size of the resource forks of all hidden file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical resource fork size of hidden files: "+Format(d.HiddenPhysicalResourceForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.23 HiddenPhysicalTotalSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical size of the both forks of all hidden file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical total size of hidden files: "+Format(d.HiddenPhysicalTotalSize/1000000,"0")+ " MB"
```

Notes: (Read only property)

21.1.24 IgnoreHiddenFolderContent as Boolean

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: If this flag is set, the Update method will not count invisible files.

Notes: (Read and Write property)

21.1.25 ItemCount as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of items counted.

Example:

```
// chose a folder
Var f as FolderItem = SpecialFolder.Music
// calculate size
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(true,0)

// show number of files
MsgBox str(d.ItemCount)
```

Notes: ItemCount=VisibleItemCount+HiddenItemCount
(Read only property)

21.1.26 LogicalDataForkSize as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical data fork size in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical data fork size of all files: "+Format(d.LogicalDataForkSize/1000000,"0")+ " MB"
```

Notes: (Read only property)

21.1.27 LogicalResourceForkSize as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical resource fork size in Bytes.

Notes: Var f as FolderItem = SpecialFolder.desktop

Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)

MsgBox "Logical resource fork size of all files: "+Format(d.LogicalResourceForkSize/1000000,"0")+ " MB"

(Read only property)

21.1.28 LogicalTotalSize as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical total size in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
```

```
MsgBox "Logical total size of all files: "+Format(d.LogicalTotalSize/1000000,"0")+ " MB"
```

Notes: (ResourceFork+DataFork)
(Read only property)

21.1.29 PhysicalDataForkSize as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical data fork size in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical data fork size of all files: "+Format(d.PhysicalDataForkSize/1000000,"0")+ " MB"
```

Notes: (Read only property)

21.1.30 PhysicalResourceForkSize as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical resource fork size in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical resource fork size of all files: "+Format(d.PhysicalResourceForkSize/1000000,"0")+ " MB"
```

Notes: (Read only property)

21.1.31 PhysicalTotalSize as UInt64

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical total file size in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical total size of all files: "+Format(d.VisiblePhysicalTotalSize/1000000,"0")+ " MB"
```

Notes: (ResourceFork+DataFork)
That's what the Finder shows you.
(Read only property)

21.1.32 QueryCompressedSizes as Boolean

Plugin Version: 11.3, Platform: Windows, Targets: All.

Function: Whether to query compressed file sizes on Windows.

Notes: (Read and Write property)

21.1.33 RecursionLimit as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The limit for recursion.

Notes: Default is -1 for no limit.

Zero means we do not recurse into subfolders.

Other values define how many recursion levels are allowed.

(Read and Write property)

21.1.34 RecursionMaxLevel as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The maximum levels of recursions we had for searching this folder.

Notes: (Read and Write property)

21.1.35 VisibleCompressedSize as UInt64

Plugin Version: 11.3, Platform: Windows, Targets: All.

Function: The size in bytes of all visible files.

Notes: If a file is compressed, we add here the compressed size.

If a file is not compressed, we add the normal logical size.

Only used on Windows and only with QueryCompressedSizes = true.

(Read only property)

21.1.36 VisibleFilesCount as Integer

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of visible files.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.VisibleFilesCount)+" visible files"
```

Notes: (Read and Write property)

21.1.37 VisibleFolderCount as Integer

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of visible folders.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.VisibleFolderCount)+" visible folders"
```

Notes: (Read and Write property)

21.1.38 VisibleItemCount as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of visible items.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox str(d.VisibleItemCount)+" visible items"
```

Notes: Items are folders or files.

VisibleItemCount=VisibleFolderCount+VisibleFilesCount
(Read only property)

21.1.39 VisibleLogicalDataForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical size of the data forks of all visible file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical data fork size of visible files: "+Format(d.VisibleLogicalDataForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.40 VisibleLogicalResourceForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical size of the resource forks of all visible file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical resource fork size of visible files: "+Format(d.VisibleLogicalResourceForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.41 VisibleLogicalTotalSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical size of the both forks of all visible file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Logical total size of visible files: "+Format(d.VisibleLogicalTotalSize/1000000,"0")+
" MB"
```

Notes: (Read only property)

21.1.42 VisiblePhysicalDataForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical size of the data forks of all visible file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical data fork size of visible files: "+Format(d.VisiblePhysicalDataForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.43 VisiblePhysicalResourceForkSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical size of the resource forks of all visible file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical resource fork size of visible files: "+Format(d.VisiblePhysicalResourceForkSize/1000000,"0")+
" MB"
```

Notes: (Read and Write property)

21.1.44 VisiblePhysicalTotalSize as UInt64

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical size of the both forks of all visible file in Bytes.

Example:

```
Var f as FolderItem = SpecialFolder.desktop
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)
MsgBox "Physical total size of visible files: "+Format(d.VisiblePhysicalTotalSize/1000000,"0")+ " MB"
```

Notes: (Read only property)

21.1.45 YieldTicks as Integer

Plugin Version: 7.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: How much time is given back to Xojo for other ticks.

Example:

```
Var d as DirectorySizeMBS // your DirectorySizeMBS object
d.YieldTicks=6 // only use 1/10th of a second
```

Notes: If value is greater than zero, the application will yield to another Xojo thread after the given number of ticks have passed. 60 ticks are one second. Using a small value can slow down processing a lot while a big value keeps your application not responding to mouse clicks.

If you use this property with e.g. 6 as the value, you may also want to use this method in a thread so you can handle mouse events or let Xojo redraw a progressbar.

(Read and Write property)

21.2 Globals

21.2.1 ExchangeFilesMBS(first as folderitem, second as folderitem) as Integer

Plugin Version: 10.4, Platforms: macOS, Windows, Targets: All.

Function: Exchanges two files.

Example:

```
Var f1 as FolderItem = SpecialFolder.Desktop.Child("some picture.jpg")
Var f2 as FolderItem = SpecialFolder.Desktop.Child("another picture.jpg")

Var e as Integer = ExchangeFilesMBS(f1,f2)
```

```
MsgBox str(e) // show error code (0=no error)
```

Notes: On Mac swaps the contents of two files:

The ExchangeFilesMBS function allows programs to implement a "safe save" operation by creating and writing a complete new file and swapping the contents. An folderitem, alias, FSSpec, or FSRef that refers to the old file will now access the new data. The corresponding information in in-memory data structures are also exchanged.

Either or both files may have open access paths. After the exchange, the access path will refer to the opposite file's data (that is, to the same data it originally referred, which is now part of the other file).

On Windows files are renamed so they exchange their paths.

PS: Xojo does not notice the change, so the folderitems you pass should no longer be used. To access the file, please make new folderitem with parent.truechild(filename).

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr3](#)
- [MBS REALbasic Plugins, version 10.6pr9](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr12](#)

21.2.2 AdminToolsMBS(domain as Integer) as folderitem

Plugin Version: 13.0, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to the admin tools folder on Windows.

Notes: Returns on Windows the common admin tools folder for domain = -32766 and user's admin tools folder if domain = -32763.

21.2.3 CookiesMBS as folderitem

Plugin Version: 13.0, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to the Cookie folder on Windows.

21.2.4 HistoryMBS as folderitem

Plugin Version: 13.0, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to the history folder on Windows.

21.2.5 InternetCacheMBS as folderitem

Plugin Version: 13.0, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to the Internet Cache folder on Windows.

21.2.6 WindowsStartMenuMBS(domain as Integer) as folderitem

Plugin Version: 13.0, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to the start menu folder on Windows.

21.2.7 SetCurrentWorkingDirectoryMBS(path as folderitem) as boolean

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the current working directory for the application.

Example:

```
MsgBox SpecialFolder.CurrentWorkingDirectory.NativePath  
call SetCurrentWorkingDirectoryMBS(SpecialFolder.Desktop)
```

MsgBox SpecialFolder.CurrentWorkingDirectory.NativePath

Notes: This is sometimes needed if some library references files relative to this current working directory. Returns true on success and false on failure.

Blog Entries

- [MBS Real Studio Plugins, version 11.3fc](#)

21.2.8 SetDllDirectoryMBS(path as folderitem) as boolean

Plugin Version: 25.1, Platform: Windows, Targets: All.

Function: Adds a directory to the search path used to locate DLLs for the application.

Notes: The directory to be added to the search path. If this parameter is an empty string (""), the call removes the current directory from the default DLL search order. If this parameter is nil, the function restores the default search order.

If the function succeeds, the return value is true, otherwise false.

The SetDllDirectory function affects all subsequent calls to the LoadLibrary and LoadLibraryEx functions. It also effectively disables safe DLL search mode while the specified directory is in the search path.

After calling SetDllDirectory, the standard DLL search path is:

- The directory from which the application loaded.
- The directory specified by the SetDllDirectory function.
- The system directory. Use the GetSystemDirectory function to get the path of this directory. The name of this directory is System32.
- The Windows directory. Use the GetWindowsDirectory function to get the path of this directory.
- The directories that are listed in the PATH environment variable.

Each time the SetDllDirectory function is called, it replaces the directory specified in the previous SetDllDirectory call.

To revert to the standard search path used by LoadLibrary and LoadLibraryEx, call SetDllDirectory with nil. This also restores safe DLL search mode based on the SafeDllSearchMode registry value.

Blog Entries

- [MBS Xojo Plugins, version 25.1pr3](#)

21.2.9 VolumeFreeSizePathMBS(Path as String) as Int64

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the size of the free space of the volume which the path points to.

Notes: See also folderitem.VolumeFreeSizeMBS function.

Returns -1 on error.

For Windows, the path should point to directory and end with a backslash.

Blog Entries

- [MBS Xojo Plugins, version 20.3pr1](#)

21.2.10 VolumeSizePathMBS(Path as String) as Int64

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the size on the volume which the path points to.

Notes: See also folderitem.VolumeSizeMBS function.

Returns -1 on error.

For Windows, the path should point to directory and end with a backslash.

Blog Entries

- [MBS Xojo Plugins, version 20.3pr1](#)

21.3 class FileListMBS

21.3.1 class FileListMBS

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a list of files.

Notes: This class is made to get a list of files in a folder faster than by using a folderitem.

Blog Entries

- [MBS Xojo Plugins, version 25.1pr1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 22.5](#)
- [MBS Xojo Plugins, version 22.5pr7](#)
- [MBS Xojo Plugins, version 21.2pr6](#)
- [MBS Xojo Plugins, version 21.2pr4](#)
- [MBS Xojo Plugins, version 20.6pr4](#)
- [MBS Xojo Plugins in version 19.0](#)

- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.1](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [NSBundleMBS and NSDirectoryEnumeratorMBS](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [17.2, page 10: News](#)

21.3.2 Methods

21.3.3 AttributeModificationDate(index as Integer) as Double

Plugin Version: 9.1, Platform: macOS, Targets: All.

Function: The attribute modification date of this file or directory.

Notes: On Mac OS the UTC date (+0 time zone)

Raises an exception if index is out of bounds. Index is zero based.

See also:

- [21.3.4 AttributeModificationDate\(index as Integer, UTC as boolean\) as Date](#) 398

21.3.4 AttributeModificationDate(index as Integer, UTC as boolean) as Date

Plugin Version: 13.4, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The attribute modification date of this file or directory.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

See also:

- [21.3.3 AttributeModificationDate\(index as Integer\) as Double](#) 398

21.3.5 AttributeModificationDateTime(index as integer, UTC as boolean) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The attribute modification date of this file or directory.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

21.3.6 BackupDate(index as Integer) as Double

Plugin Version: 9.1, Platform: macOS, Targets: All.

Function: The backup date of this file or directory.

Notes: On Mac OS the UTC date (+0 time zone)

See also:

- 21.3.7 BackupDate(index as Integer, UTC as boolean) as Date

399

21.3.7 BackupDate(index as Integer, UTC as boolean) as Date

Plugin Version: 13.4, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The backup date of this file or directory.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

See also:

- 21.3.6 BackupDate(index as Integer) as Double

399

21.3.8 BackupDateTime(index as integer, UTC as boolean) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The backup date of this file or directory.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

21.3.9 CFURL(index as integer) as Variant

Plugin Version: 19.0, Platform: macOS, Targets: All.

Function: Queries CFURLMBS object for an entry.

Notes: Returns nil if no CFURL is kept for this item.

Raises an exception if index is out of bounds. Index is zero based.

21.3.10 Close

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

Notes: There is no need to call this method except you want to free all resources of this object now without

waiting for Xojo to do it for you.

21.3.11 Constructor

Plugin Version: 5.4, Platforms: macOS, Linux, Targets: All.

Function: A dummy constructor used only for automatic plugin testing.

See also:

- 21.3.12 Constructor(filelist as FileListMBS, index as Integer, WinFilter as string = "", SkipMode as Integer = 0) 400
- 21.3.13 Constructor(folder as folderitem, WinFilter as string = "", SkipMode as Integer = 0) 401
- 21.3.14 Constructor(Path as String, WinFilter as string = "", SkipMode as Integer = 0) 401

21.3.12 Constructor(filelist as FileListMBS, index as Integer, WinFilter as string = "", SkipMode as Integer = 0)

Plugin Version: 6.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor which creates a new file list based on item in a given file list.

Example:

```

Var f1 as new FileListMBS(SpecialFolder.UserHome)
'MsgBox f1.path

// search first index of a visible folder
Var IndexOfDirectory as integer = 0
while f1.Directory(IndexOfDirectory) = false or f1.Visible(IndexOfDirectory) = false
  IndexOfDirectory = IndexOfDirectory + 1
wend

// list that folder
Var f2 as new FileListMBS(f1, IndexOfDirectory)
'MsgBox f2.path

// show first file and path
MsgBox f2.Name(0)+EndOfLine+f2.ItemPath(0)

```

Notes: If count is 0 after the constructor the folder is invalid or empty.

On Windows the WinFilter allows you to pass a custom filter like "*.txt" to only find some files there.

SkipMode, added in version 20.1, allows you to skip some files from being part of the file list. Please use BitwiseOr() to combine Skip flags, e.g. BitwiseOr(SkipHidden, SkipFolders) returns only visible files.

See also:

21.3. CLASS FILELISTMBS	401
• 21.3.11 Constructor	400
• 21.3.13 Constructor(folder as folderitem, WinFilter as string = "", SkipMode as Integer = 0)	401
• 21.3.14 Constructor(Path as String, WinFilter as string = "", SkipMode as Integer = 0)	401

21.3.13 Constructor(folder as folderitem, WinFilter as string = "", SkipMode as Integer = 0)

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor which creates a new file list based on the folder.

Notes: If count is 0 after the constructor the folder is invalid or empty.

On Windows the WinFilter allows you to pass a custom filter like "*.txt" to only find some files there.

SkipMode, added in version 20.1, allows you to skip some files from being part of the file list. Please use BitwiseOr() to combine Skip flags, e.g. BitwiseOr(SkipHidden, SkipFolders) returns only visible files.

See also:

- 21.3.11 Constructor 400
- 21.3.12 Constructor(filelist as FileListMBS, index as Integer, WinFilter as string = "", SkipMode as Integer = 0) 400
- 21.3.14 Constructor(Path as String, WinFilter as string = "", SkipMode as Integer = 0) 401

21.3.14 Constructor(Path as String, WinFilter as string = "", SkipMode as Integer = 0)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor which creates a new file list based on the directory path.

Notes: If count is 0 after the constructor the directory path is invalid or empty.

On Windows the WinFilter allows you to pass a custom filter like "*.txt" to only find some files there.

SkipMode, added in version 20.1, allows you to skip some files from being part of the file list. Please use BitwiseOr() to combine Skip flags, e.g. BitwiseOr(SkipHidden, SkipFolders) returns only visible files.

See also:

- 21.3.11 Constructor 400
- 21.3.12 Constructor(filelist as FileListMBS, index as Integer, WinFilter as string = "", SkipMode as Integer = 0) 400
- 21.3.13 Constructor(folder as folderitem, WinFilter as string = "", SkipMode as Integer = 0) 401

21.3.15 `CreationDate(index as Integer)` as Double

Plugin Version: 5.1, Platforms: macOS, Windows, Targets: All.

Function: The creation date.

Notes: On Mac OS the UTC date (+0 time zone)

On Windows looks like local timezone.

See also:

- 21.3.16 `CreationDate(index as Integer, UTC as boolean)` as Date 402

21.3.16 `CreationDate(index as Integer, UTC as boolean)` as Date

Plugin Version: 13.4, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The creation date.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

See also:

- 21.3.15 `CreationDate(index as Integer)` as Double 402

21.3.17 `CreationDateTime(index as integer, UTC as boolean)` as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The creation date.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

21.3.18 `Creator(index as Integer)` as string

Plugin Version: 5.1, Platforms: macOS, Linux, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The Mac Creator code of the item with the given index.

Notes: Always "" on Windows and Linux.

Raises an exception if index is out of bounds. Index is zero based.

Stopped working on macOS Monterey and newer since Apple deprecated the API long ago.

21.3.19 Directory(index as Integer) as boolean

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the item with the given index is a folder.

Notes: True if it is a folder and false if it is a file.

Raises an exception if index is out of bounds. Index is zero based.

21.3.20 DisplayName(index as Integer) as string

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The display name of the item with the given index.

Notes: Asks the System for the display name of the item. Returns the normal name on any error.

As this call may cost quite some CPU time you may cache the value if you need it more often.

May return a Unicode string, so be carefull with encoding.

Returns "" on any error.

Raises an exception if index is out of bounds. Index is zero based.

21.3.21 FinderFlags(index as Integer) as Integer

Plugin Version: 6.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Finder flags from Mac OS.

Notes: On Windows and Linux value is 0.

The flag value is a set of bits with the following meaning:

21.3.22 FSRef(index as Integer) as memoryblock

Plugin Version: 5.2, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The FSRef used for this item.

Notes: Returns nil if no FSRef is kept for this item.

On Mac OS there is a FSSpec or (FSRef and HFSUniStr255).

Raises an exception if index is out of bounds. Index is zero based.

bit 15	value &H8000	isAlias
bit 14	value &H4000	isInvisible
bit 13	value &H2000	hasBundle (has a BNDL resource)
bit 12	value &H1000	nameLocked
bit 11	value &H0800	isStationary
bit 10	value &H0400	hasCustomIcon
bit 8	value &H0100	hasBeenInitd (Finder has seen the file since it has been created)
bit 7	value &H0080	hasNoINITs (there is no INIT rsrc in the Extension file)
bit 6	value &H0040	isShared
bits 1-3	value &H000E	color (as a 3-bit value from 0-7)

21.3.23 HFSUniStr255(index as Integer) as memoryblock

Plugin Version: 5.2, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The HFSUniStr255 used for this item.

Notes: Returns nil if no HFSUniStr255 is kept for this item.

On Mac OS there is a FSSpec or (FSRef and HFSUniStr255).

Raises an exception if index is out of bounds. Index is zero based.

21.3.24 IsBundle(index as Integer) as Boolean

Plugin Version: 6.1, Platforms: macOS, Windows, Targets: All.

Function: Queries using LaunchServices whether this item is a folder and the root of a bundle.

Example:

```
Protected Sub Dir(f as FolderItem)
```

```
Var l as FileListMBS
```

```
Var i,c as Integer
```

```
// search subfolders and find bundles
```

```
l=new FileListMBS(f)
```

```
c=l.Count-1
```

```
for i=0 to c
```

```
// if bundle, show it in listbox
```

```
if l.IsBundle(i) then
```

```
Listbox1.AddRow l.Item(i).NativePath
```

```
elseif l.Directory(i) then
```

```
// if sub folder, browse it
```

```
dir l.Item(i)
```

```
end if
```

```
next
```

End Sub

Notes: On Windows, Linux and Mac OS Classic always false.

21.3.25 IsHardLinked(index as Integer) as boolean

Plugin Version: 9.4, Platforms: macOS, Linux, Targets: All.

Function: Whether the file is a hard link.

Notes: If two directory entries point to the same hard disc space, you have a file which has two directory entries, but only one storage.

This function returns true for files in a Time Machine backup which did not change since the last backup and share their disc space with the other backups.

Raises an exception if index is out of bounds. Index is zero based.

21.3.26 Item(index as Integer) as folderitem

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a folderitem for the item with the given index.

Notes: Returns nil on any error.

The reason why you use this class is to avoid making folderitems which is slow, so don't use this function too often.

Raises an exception if index is out of bounds. Index is zero based.

21.3.27 ItemPath(index as Integer) as string

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The native file path for an item.

21.3.28 LastAccessDate(index as Integer) as Double

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last access date of this file or directory.

Notes: On Mac OS the UTC date (+0 time zone)

Raises an exception if index is out of bounds. Index is zero based.

To query Spotlight's Last Open day, please check the FAQ for sampel code.

See also:

- 21.3.29 `LastAccessDate(index as Integer, UTC as boolean) as Date` 406

21.3.29 `LastAccessDate(index as Integer, UTC as boolean) as Date`

Plugin Version: 13.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The last access date of this file or directory.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

To query Spotlight's Last Open day, please check the FAQ for sample code.

See also:

- 21.3.28 `LastAccessDate(index as Integer) as Double` 405

21.3.30 `LastAccessDateTime(index as integer, UTC as boolean) as DateTime`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last access date of this file or directory.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

To query Spotlight's Last Open day, please check the FAQ for sample code.

21.3.31 `LogicalDataLength(index as Integer) as Int64`

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical file data length for the item with the given index.

Notes: Same value as `folderitem.length`, but works with files >2GB.

Raises an exception if index is out of bounds. Index is zero based.

21.3.32 `LogicalResourceLength(index as Integer) as Int64`

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The logical resource length of the item with the given index.

Notes: Raises an exception if index is out of bounds. Index is zero based.

21.3.33 ModificationDate(index as Integer) as Double

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The modification date.

Notes: On Mac OS the UTC date (+0 time zone)

On Windows looks like local timezone.

Raises an exception if index is out of bounds. Index is zero based.

See also:

- 21.3.34 ModificationDate(index as Integer, UTC as boolean) as Date

407

21.3.34 ModificationDate(index as Integer, UTC as boolean) as Date

Plugin Version: 13.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The modification date.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

See also:

- 21.3.33 ModificationDate(index as Integer) as Double

407

21.3.35 ModificationDateTime(index as integer, UTC as boolean) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The modification date.

Notes: Either in UTC time zone or in local time zone.

Raises an exception if index is out of bounds. Index is zero based.

21.3.36 Name(index as Integer) as string

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the item with the given index.

Notes: May return a Unicode string, so be carefull with encoding.

Returns "" on any error.

Raises an exception if index is out of bounds. Index is zero based.

21.3.37 NodeID(index as Integer) as Int64

Plugin Version: 9.1, Platforms: macOS, Linux, Targets: All.

Function: The Mac node ID for the given file or directory.

Notes: The node id is zero on Windows and Linux.

Raises an exception if index is out of bounds. Index is zero based.

21.3.38 ParentDirectoryID(index as Integer) as Int64

Plugin Version: 9.1, Platforms: macOS, Linux, Targets: All.

Function: The Mac ID for the parent directory.

Notes: Parent Directory ID is zero on Windows and Linux.

Raises an exception if index is out of bounds. Index is zero based.

21.3.39 PhysicalDataLength(index as Integer) as Int64

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical data length of the item with the given index.

Notes: Returns 0 on any error.

Raises an exception if index is out of bounds. Index is zero based.

21.3.40 PhysicalResourceLength(index as Integer) as Int64

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The physical resource length of the item with the given index.

Notes: Value is always 0 on Windows.

Returns 0 on any error.

Raises an exception if index is out of bounds. Index is zero based.

21.3.41 SortByCreationDate

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sorts items in list by creation date.

21.3.42 SortByFileName

Plugin Version: 19.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sorts items in list by file names.

21.3.43 SortByModificationDate

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sorts items in list by modification date.

21.3.44 TrueItem(index as Integer) as folderitem

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a folderitem for the item with the given index.

Notes: Returns nil on any error.

The reason why you use this class is to avoid making folderitems which is slow, so don't use this function too often. e.g. good if you just take 5 folderitems for 20000 files by filtering.

TrueItem will use GetTrueFolderItem so the link/alias files are not resolved by Xojo.

Due to bugs in Xojo, you may still get alias resolved.

21.3.45 Type(index as Integer) as string

Plugin Version: 5.1, Platforms: macOS, Linux, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The Mac Type code of the item with the given index.

Notes: Always "" on Windows and Linux.

Raises an exception if index is out of bounds. Index is zero based.

Stopped working on macOS Monterey and newer since Apple deprecated the API long ago.

21.3.46 Visible(index as Integer) as boolean

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the item with the given index is visible.

Notes: On Mac OS X, files with name starting with "." are considered to be invisible.

False on any error.

Raises an exception if index is out of bounds. Index is zero based.

21.3.47 WinFileAttributes(index as Integer) as Integer

Plugin Version: 6.1, Platforms: macOS, Windows, Targets: All.

Function: The file attributes for this file on Windows.

Notes: Value is 0 on other platforms.

Raises an exception if index is out of bounds. Index is zero based.

21.3.48 Properties

21.3.49 Cancel as Boolean

Plugin Version: 22.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tells FileListMBS to stop gathering files.

Notes: To be used with threaded flag to stop background threads from working on directory listings soon. You may want to set FileListMBS.cancel to true in app.CancelClose when you let the application has a running thread, so you can quit properly.

The constructors set cancel = false to make sure they work.
(Read and Write property)

21.3.50 Count as Integer

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of items found.

Notes: So index goes from 0 to count-1 in all functions.

(Read and Write property)

21.3.51 Folder as FolderItem

Plugin Version: 5.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The folderitem used in the constructor.

Notes: (Read and Write property)

21.3.52 OK as Boolean

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the process of reading the directory content in the constructor was successful.

Notes: Should be true on success.

(Read and Write property)

21.3.53 Path as String

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The file path for the folder.

Notes: The plugin makes sure it ends with slash or backslash.

(Read and Write property)

21.3.54 Threaded as Boolean

Plugin Version: 22.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether you want to run FileListMBS constructors threaded.

Notes: If you run code in a Xojo thread, which uses FileListMBS, you can set this property to true. We then do the work on a preemptive thread to keep the GUI responsive.

(Read and Write property)

21.3.55 TotalLogicalDataLength as Int64

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The sum of logical data length for all items.

Notes: (Read only property)

21.3.56 TotalLogicalResourceLength as Int64

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The sum of logical resource length for all items.

Notes: (Read only property)

21.3.57 TotalPhysicalDataLength as Int64

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The sum of physical data length for all items.

Notes: (Read only property)

21.3.58 TotalPhysicalResourceLength as Int64

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The sum of physical resource length for all items.

Notes: (Read only property)

21.3.59 YieldTicks as Integer

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: How much time is given back to Xojo for other ticks.

Example:

```
FileListMBS.YieldTicks = 6
```

Notes: If value is greater than zero, the application will yield to another Xojo thread after the given number of ticks have passed. 60 ticks are one second. Using a small value can slow down processing a lot while a big value keeps your application not responding to mouse clicks.

If you use this property with e.g. 6 as the value, you may also want to use this method in a thread so you can handle mouse events or let Xojo redraw a progressbar.

For threaded FileListMBS, this property has no effect.

(Read and Write property)

21.3.60 Constants

Skip Modes

Constant	Value	Description
SkipFiles	1	Skip files.
SkipFolders	2	Skip folders.
SkipHidden	12	Skip hidden files and folders.
SkipHiddenFiles	4	Skip hidden files.
SkipHiddenFolders	8	Skip hidden folders.
SkipNone	0	Skip no files.

21.4 class FolderItem

21.4.1 class FolderItem

Platforms: macOS, Linux, Windows, Targets: All.

Function: One of Xojo's base classes.

Notes: Handles access to files.

21.4.2 Methods

21.4.3 CalculateDirectorySizeMBS(recursive as boolean = false, ticks as Integer = 0, QueryCompressedSizes as boolean = false, RecursionLimit as Integer = -1) as DirectorySizeMBS

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns an object with informations about the folder size.

Example:

```
// chose a folder
Var f as FolderItem = SelectFolder

// calculate
Var d as DirectorySizeMBS = f.CalculateDirectorySizeMBS(True,0)

// display
MsgBox str(d.FilesCount)+" files in "+str(d.FolderCount)+" folder"
```

Notes: recursive: Whether to count items in the subfolders.

ticks: Whether to yield time to other threads. (See YieldTicks property)

QueryCompressedSizes: Whether to query compressed file sizes on Windows.

RecursionLimit: If positive, the recursion limit.

Returns nil on any error.

Blog Entries

- [MBS Real Studio Plugins, version 12.4pr9](#)

21.4.4 CompressedFileLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the compressed file on disk.

Example:

```
filesize.text=format(file.CompressedFileLengthMBS,"0")
```

Notes: On Windows files can be compressed and this function returns the size of the compressed file. On Mac and for non compressed files on Windows this function returns the uncompressed size.

21.4.5 CreateLargeBinaryStreamMBS(MacType as string, MacCreator as string) as LargeBinaryStreamMBS

Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a file as a LargeBinaryStreamMBS.

Example:

```
Var f as FolderItem // your file
Var l as LargeBinaryStreamMBS
```

```
l=f.CreateLargeBinaryStreamMBS("TEXT","ttx")
```

Notes: If there is already a file, it is deleted.

On Windows the parameters are ignored.

Returns nil on any error.

21.4.6 CreateResStreamMBS(MacType as string, MacCreator as string) as ResStreamMBS

Platform: macOS, Targets: All.

Function: Creates a new ResStreamMBS.

Notes: If there is already a file, it is deleted.

If the file could not be created it is deleted.

Returns nil on any error.

21.4.7 DeleteDataForkMBS

Platform: macOS, Targets: All.

Function: Deletes the data fork of a file.

Notes: Equal to open the file using a binarystream and setting the length property to 0.

On Mac OS a file can exist without a datafork, with a resource fork or even without any fork.

Blog Entries

- [MBS Plugins 10.3 Release Notes](#)

21.4.8 DeleteResourceForkMBS

Platform: macOS, Targets: All.

Function: Deletes the resource fork of a file.

Notes: Equal to open the file using a ResStreamMBS and setting the length property to 0.

On Mac OS a file can exist without a datafork, with a resource fork or even without any fork.

Blog Entries

- [MBS Plugins 10.3 Release Notes](#)

21.4.9 DisplayPathMBS(delimiter as string = "/") as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shows display path.

Example:

// Sample values in German:

// SnowLeopard/Programme/Dienstprogramme/Konsole

```
MsgBox SpecialFolder.Applications.Child("Utilities").Child("Console.app").DisplayPathMBS("/")
```

// SnowLeopard/Benutzer/cs/Schreibtisch

```
MsgBox SpecialFolder.Desktop.DisplayPathMBS("/")
```

// SnowLeopard ,û Programme ,û Dienstprogramme ,û Konsole

```
MsgBox SpecialFolder.Applications.Child("Utilities").Child("Console.app").DisplayPathMBS(" ,û ")
```

// SnowLeopard ,û Benutzer ,û cs ,û Schreibtisch

```
MsgBox SpecialFolder.Desktop.DisplayPathMBS(" ,û ")
```

Notes: You can pass any string for delimiter. Good looks an arrow or just a slash.

Blog Entries

- [MBS Xojo Plugins, version 17.5pr3](#)

21.4.10 FilesMBS as FolderItem()

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of files in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var files() as FolderItem = folder.FilesMBS
MsgBox str(UBound(files)+1)+" files"
```

Notes: Similar to item() function, but returns all files with one call.

Blog Entries

- [MBS Real Studio Plugins, version 12.0fc1](#)

21.4.11 FoldersMBS as FolderItem()

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of folders in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var folders() as FolderItem = folder.FoldersMBS
MsgBox str(UBound(folders)+1)+" folders"
```

Notes: Similar to item() function, but returns all folders with one call.

Blog Entries

- [MBS Real Studio Plugins, version 12.2pr2](#)
- [MBS Real Studio Plugins, version 12.0fc1](#)

21.4.12 FontActivateMBS(OnlyLocal as boolean) as Integer

Plugin Version: 4.3, Platforms: macOS, Windows, Targets: All.

Function: Activates one font.

Example:

```
Var f as folderitem
Var e as Integer

f=SpecialFolder.Desktop.child("MyWonderfulFont.dfont")
e=f.FontActivateMBS(true)

select case e
case 0
msgbox "No error"
case -1
```



```
MsgBox "Parameter error. (Plugin)"
case 1 // Windows error
MsgBox "Parameter error. (System)"
case -43 // Mac error
MsgBox "File not found."
case -50 // Mac error
MsgBox "Parameter error. (System)"
case -108 // Mac error
MsgBox "File is not a font file."
else
MsgBox "Error: "+str(e)
end Select
```

Notes: The folderitem used here points to a font file.

If you want the Font Manager to make fonts not visible to all applications installed on the system, use the OnlyLocal property.

Requires Mac OS 9.0 or newer.

Returns -1 if this FontManager function was not found. 0 on success.

Some error codes for Mac:

0	No error.
-43	File not found.
-45	File locked.
-108	Out of memory.
-105	Already registered.

Fails in Xojo 2017 on Windows due to changes in Xojo's way to handle fonts. The plugin still activates them, but Xojo doesn't allow you to use fonts which haven't been there earlier. See feedback case 46596.

Blog Entries

- [MBS Xojo Plugins, version 23.4pr3](#)
- [MBS Xojo / Real Studio Plugins, version 14.4pr6](#)
- [MBS Real Studio Plugins, version 12.1pr5](#)
- [MBS REALbasic Plugins, version 10.5pr2](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

Xojo Developer Magazine

- [10.1, pages 75 to 76: Using Plugins, Working with the Monkeybread Plugins by Marc Zeedar](#)

21.4.13 FontDeactivateMBS(OnlyLocal as boolean) as Integer

Plugin Version: 4.3, Platforms: macOS, Windows, Targets: All.

Function: Deactivates one font.

Notes: The folderitem used here points to font file.

The OnlyLocal must match the value which was used on registration.

Requires Mac OS 9.0 or newer, Windows 2000/XP.

Returns -1 if this FontManager function was not found. 0 on success.

An user reported that this does not work on Mac OS X 10.4.

Some error codes for Mac:

0	No error.
-43	File not found.
-45	File locked.
-108	Out of memory.

On Windows you may need to call this function several times until all references of the file are freed.

Blog Entries

- [MBS Real Studio Plugins, version 12.1pr5](#)
- [MBS Real Studio Plugins, version 11.2pr3](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

21.4.14 IsCompressedFileMBS as Boolean

Platform: Windows, Targets: All.

Function: Is this file a compressed one?

Notes: On Windows you can compress individual files to double your harddisk, but no application may note, because this compression is transparent.

21.4.15 IsEncryptedFileMBS as Boolean

Platform: Windows, Targets: All.

Function: Is this file an encrypted one?

Notes: On Windows you can encrypted individual files to secure your harddisk, but no application may note, because this encryption is transparent to them.

21.4.16 IsFileDataForkOpenReadWriteMBS as boolean

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test whether a file is currently opened for read and write by an application.

Notes: This method tries to open the file with exclusive access. If this fails for access/permission errors, the function returns true.

Blog Entries

- [MBS Plugins updated for Xojo 2019r2](#)

21.4.17 IsFileResourceForkOpenReadWriteMBS as boolean

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test whether a file is currently opened for read and write by an application.

Notes: This method tries to open the file with exclusive access. If this fails for access/permission errors, the function returns true.

On Windows, the function result is always false.

Blog Entries

- [MBS Plugins updated for Xojo 2019r2](#)

21.4.18 ItemsMBS as FolderItem()

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of items in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var Items() as FolderItem = folder.ItemsMBS
MsgBox str(UBound(Items)+1)+" items"
```

Notes: Similar to item() function, but returns all items with one call.

Blog Entries

- [MBS Xojo Plugins, version 20.0pr6](#)
- [MBS Real Studio Plugins, version 12.0fc1](#)

21.4.19 LogicalFileDataLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the logical disk space used for this file's data fork.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
```

```
Var lines(-1) as string
```

```
lines.Append "Length: "+str(F.Length)
lines.Append "ResourceForkLength: "+str(F.ResourceForkLength)
lines.Append "LogicalFileDataLengthMBS: "+str(F.LogicalFileDataLengthMBS)
lines.Append "LogicalFileResLengthMBS: "+str(F.LogicalFileResLengthMBS)
lines.Append "LogicalFileTotalLengthMBS: "+str(F.LogicalFileTotalLengthMBS)
lines.Append "PhysicalFileDataLengthMBS: "+str(F.PhysicalFileDataLengthMBS)
lines.Append "PhysicalFileResLengthMBS: "+str(F.PhysicalFileResLengthMBS)
lines.Append "PhysicalFileTotalLengthMBS: "+str(F.PhysicalFileTotalLengthMBS)
```

```
MsgBox Join(lines,EndOfLine)
```

Notes: This function works for files bigger than 2 GB which RB's built in functions don't.

On Windows the physical size reported is equal to the logical size, because there is no function for the physical size.

21.4.20 LogicalFileResLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the logical disk space used for this file's resourcefork.

Example:

```
filesize.text=format(file.LogicalFileResLengthMBS,"0")
```

Notes: This function works for files bigger than 2 GB which RB's built in functions don't.

On Windows the physical size reported is equal to the logical size, because there is no function for the physical size.

21.4.21 LogicalFileTotalLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the logical disk space used for this file's datafork.

Example:

```
filesize.text=format(file.LogicalFileTotalLengthMBS,"0")
```

Notes: This function works for files bigger than 2 GB which RB's built in functions don't.

On Windows the physical size reported is equal to the logical size, because there is no function for the physical size.

21.4.22 NameExtensionMBS as string

Plugin Version: 10.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The file extension from the file name.

Example:

```
Var f as folderitem = SpecialFolder.desktop
```

```
MsgBox "Name: "+f.Name+EndOfLine+"Name extension: "+f.NameExtensionMBS+EndOfLine+"Name  
without extension: "+f.NameWithoutExtensionMBS
```

```
Var g as FolderItem = SpecialFolder.Desktop.Child("test.rbp")
```

```
MsgBox "Name: "+g.Name+EndOfLine+"Name extension: "+g.NameExtensionMBS+EndOfLine+"Name  
without extension: "+g.NameWithoutExtensionMBS
```

Notes: If there is no extension, this string is empty.
The extension does not include the dot.

Blog Entries

- [MBS Real Studio Plugins, version 13.1pr17](#)
- [MBS Plugins 10.3 Release Notes](#)
- [MBS REALbasic Plugins, version 10.3pr8](#)

Xojo Developer Magazine

- [20.1, page 65: Send Your Emails, How to use the MBS Xojo plugins to send emails by Stefanie Juchmes](#)

21.4.23 NameWithoutExtensionMBS as string

Plugin Version: 10.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The file name without file extension.

Example:

```
Var f as folderitem = SpecialFolder.desktop
```

```
MsgBox "Name: "+f.Name+EndOfLine+"Name extension: "+f.NameExtensionMBS+EndOfLine+"Name  
without extension: "+f.NameWithoutExtensionMBS
```

```
Var g as FolderItem = SpecialFolder.Desktop.Child("test.rbp")
```

```
MsgBox "Name: "+g.Name+EndOfLine+"Name extension: "+g.NameExtensionMBS+EndOfLine+"Name  
without extension: "+g.NameWithoutExtensionMBS
```

Notes: If there is no extension, the whole name is returned.

Blog Entries

- [MBS Real Studio Plugins, version 13.1pr17](#)
- [MBS Plugins 10.3 Release Notes](#)
- [MBS REALbasic Plugins, version 10.3pr8](#)

Xojo Developer Magazine

- [20.1, page 65: Send Your Emails, How to use the MBS Xojo plugins to send emails by Stefanie Juchmes](#)

21.4.24 OpenAsLargeBinaryStreamMBS(write as Boolean) as LargeBinaryStreamMBS

Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens a file as a LargeBinaryStreamMBS.

Example:

```
Var l as LargeBinaryStreamMBS
```

```
Var f as FolderItem = SpecialFolder.Desktop.Child("testfile")
```

```
l=f.OpenAsLargeBinaryStreamMBS(true)
```

Notes: Returns nil on any error.

21.4.25 OpenAsResStreamMBS(write as Boolean) as ResStreamMBS

Platform: macOS, Targets: All.

Function: Opens a file's resourcefork as a ResStreamMBS.

Notes: Returns nil on any error.

21.4.26 ParentVolumeMBS as folderitem

Plugin Version: 9.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the folderitem for the volume the folderitem is pointing to.

Blog Entries

- [MBS Xojo Plugins, version 19.5pr7](#)

21.4.27 PhysicalFileDataLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the physical disk space used for this file's data fork.

Example:

```
filesize.text=format(file.PhysicalFileDataLengthMBS,"0")
```

Notes: This function works for files bigger than 2 GB which RB's built in functions don't.

On Windows the physical size reported is equal to the logical size, because there is no function for the physical size.

21.4.28 PhysicalFileResLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the physical disk space used for this file's resourcefork.

Example:

```
filesize.text=format(file.PhysicalFileResLengthMBS,"0")
```

Notes: This function works for files bigger than 2 GB which RB's built in functions don't.

On Windows the physical size reported is equal to the logical size, because there is no function for the physical size.

21.4.29 PhysicalFileTotalLengthMBS as int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the length of the physical disk space used for this file.

Example:

```
filesize.text=format(file.PhysicalFileTotalLengthMBS,"0")
```

Notes: This function works for files bigger than 2 GB which RB's built in functions don't.

On Windows the physical size reported is equal to the logical size, because there is no function for the physical size.

Blog Entries

- [MBS Xojo Plugins, version 23.5pr1](#)

21.4.30 ReadFileMBS(byref data as MemoryBlock, offset As Integer = 0, byte-Count As Integer = -1) as boolean

Plugin Version: 21.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a file to a memory block.

Example:

```
Var f As New FolderItem("/Users/cs/Downloads/Download Archive/Xcode_13_beta_5.xip", FolderItem.Path-
Modes.Native)
```

```
Var m As MemoryBlock
'Dim m As New MemoryBlock(f.Length)
```

```
If f.ReadFileMBS(m) Then
// okay
Var Len As Integer = m.size
Break
Else
// failed
Var e As Integer = f.LastErrorCode
Break
End If
```

Notes: Reads content of file and returns it in data parameter.
Returns true on success of false on failure.

See also WriteFileMBS function.

If the MemoryBlock is nil, we create one.

If the MemoryBlock is not nil and has right size, we use it, otherwise we create a new one.
Not intended for files >2 GB.

Added offset and length parameters in version 23.1.

If byteCount is negative, we read whole file. If byteCount is bigger than file, we only read the available portion.

See also:

- 21.4.31 ReadFileMBS(byref data as string, offset As Integer = 0, byteCount As Integer = -1) as boolean
425

21.4.31 ReadFileMBS(byref data as string, offset As Integer = 0, byteCount As Integer = -1) as boolean

Plugin Version: 19.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a file.

Notes: Reads content of file and returns it in data parameter.

Returns true on success of false on failure.

See also WriteFileMBS function.

Limited to maximum size of a string (2 GB)

Data must fit in memory, so 32-bit processes may be limited to reading less than 2 GB.

Added offset and length parameters in version 23.1.

If byteCount is negative, we read whole file. If byteCount is bigger than file, we only read the available portion.

Blog Entries

- [MBS Xojo Plugins, version 25.2pr4](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.1](#)
- [MBS Xojo Plugins, version 23.1pr6](#)
- [News from the MBS Xojo Plugins Version 21.4](#)
- [MBS Xojo Plugins, version 21.4pr5](#)
- [MBS Xojo Plugins in version 19.0](#)
- [MBS Xojo Plugins, version 18.6pr1](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [21.1, page 28: News from MBS Xojo Plugins, What's up with MonkeyBread Software by Stefanie Juchmes](#)
- [17.5, page 40: What's New in the MBS Plugins, With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes](#)
- [17.2, page 10: News](#)

See also:

- [21.4.30 ReadFileMBS\(byref data as MemoryBlock, offset As Integer = 0, byteCount As Integer = -1\) as boolean](#) 424

21.4.32 SortedFilesMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of files in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var Items() as FolderItem = folder.SortedFilesMBS
MsgBox str(UBound(Items)+1)+" items"
```

Notes: Checks for whether item is not a directory and returns only those.

We can resolve alias if you like and optionally sort by display name instead of normal name field.

And we can filter to only return visible items.

Blog Entries

- [Crop a two side page document to a single page document](#)
- [New in the MBS Xojo Plugins 20.0](#)
- [MBS Xojo Plugins, version 20.0pr6](#)

21.4.33 SortedFoldersMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of folders in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var Items() as FolderItem = folder.SortedFoldersMBS
```

```
MsgBox str(UBound(Items)+1)+" items"
```

Notes: Similar to `item()` function, but returns all items with one call.

Checks for whether item is directory and returns only those.

We can resolve alias if you like and optionally sort by display name instead of normal name field.

And we can filter to only return visible items.

Blog Entries

- [New in the MBS Xojo Plugins 20.0](#)
- [MBS Xojo Plugins, version 20.0pr6](#)

21.4.34 SortedItemsMBS(followAlias as Boolean = false, SortByDisplayName as boolean = false, VisibleOnly as boolean = false) as FolderItem()

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of items in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var Items() as FolderItem = folder.SortedItemsMBS
MsgBox str(UBound(Items)+1)+" items"
```

Notes: Similar to `item()` function, but returns all items with one call.

We can resolve alias if you like and optionally sort by display name instead of normal name field.

And we can filter to only return visible items.

Blog Entries

- [New in the MBS Xojo Plugins 20.0](#)
- [MBS Xojo Plugins, version 20.0pr6](#)

21.4.35 TrueFilesMBS as FolderItem()

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of files in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var files() as FolderItem = folder.TrueFilesMBS
MsgBox str(UBound(files)+1)+" files"
```

Notes: Similar to `trueitem()` function, but returns all files with one call.

Blog Entries

- [MBS Real Studio Plugins, version 12.0fc1](#)

21.4.36 TrueFoldersMBS as FolderItem()

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of folders in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var folders() as FolderItem = folder.TrueFoldersMBS
MsgBox str(UBound(folders)+1)+" folders"
```

Notes: Similar to trueitem() function, but returns all folders with one call.

Blog Entries

- [MBS Real Studio Plugins, version 12.0fc1](#)

21.4.37 TrueItemsMBS as FolderItem()

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns array of items in the given folder.

Example:

```
Var folder as FolderItem = SpecialFolder.Desktop
Var Items() as FolderItem = folder.TrueItemsMBS
MsgBox str(UBound(Items)+1)+" items"
```

Notes: Similar to trueitem() function, but returns all items with one call.

Blog Entries

- [MBS Xojo Plugins, version 25.1pr1](#)
- [MBS Real Studio Plugins, version 12.0fc1](#)

21.4.38 VolumeFreeSizeKBMBS as Int64

Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use VolumeFreeSizeMBS instead. **Function:** Returns the size on the free space on volume which the folderitem objects points to.

Example:

```
msgBox "The volume with your system folder has "+str(specialfolder.system.volumeFreeSizeKBMBS)+"
KBytes free."
```

Notes: This can't work with anything bigger than 2048 Gigabytes (until version 10.0).
Return value changed in version 10.0 from integer to SInt64.

Added Linux support in version 16.0.

21.4.39 VolumeFreeSizeMBS as Int64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the size of the free space of the volume which the folderitem objects points to.

Example:

```
// 1. a short:
msgBox "The volume with your system folder has "+str(specialfolder.system.VolumeFreeSizeMBS)+" Bytes
free."
// 2. a nicer:
Var d as Double
Var s as string

d=specialfolder.system.VolumeFreeSizeMBS

if d>10000.0 then
if d>10000000.0 then
if d>10000000000.0 then
s=format(d/1024.0/1024.0/1024.0,"0")+ " GigaBytes"
else
s=format(d/1024.0/1024.0,"0")+ " MegaBytes"
end if
else
s=format(d/1024.0,"0")+ " KiloBytes"
end if
else
s=format(d,"0")+ " Bytes"
end if

msgBox "On your drive with the system folder you have "+s+" free."
```

Notes: This should be used to handle any volume size.
Return value changed in version 10.0 from double to SInt64.

Added Linux support in version 16.0.

Returns -1 on error.

Blog Entries

- [MBS Xojo Plugins, version 20.3pr1](#)

- [MBS Xojo / Real Studio Plugins, version 16.0pr7](#)
- [MBS Xojo / Real Studio Plugins, version 14.0pr1](#)

Xojo Developer Magazine

- [2.1, page 42: Details about disks, Getting the name, the size, and more information about a disc.](#) by Christian Schmitz

21.4.40 VolumeSizeKBMBS as Int64

Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use VolumeSizeMBS instead.

Function: Returns the Size on the volume which the folderitem objects points to.

Example:

```
msgBox "The volume with your system folder is "+str(specialfolder.system.volumeSizeKBMBS)+" KB big."
```

Notes: This can't work with anything bigger than 2048 Gigabytes (until plugin version 10.0). Return value changed in version 10.0 from integer to SInt64.

Added Linux support in version 16.0.

Blog Entries

- [MBS Xojo Plugins, version 19.4pr1](#)

21.4.41 VolumeSizeMBS as Int64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the size on the volume which the folderitem objects points to.

Example:

```
// 1. a short:
msgBox "The volume with your system folder is "+str(specialfolder.system.volumeSizeMBS)+" big."
// 2. a nicer:
Var d as Double
Var s as string
```

```
d=SpecialFolder.System.volumesizeMBS
```

```
if d>10000.0 then
if d>10000000.0 then
if d>10000000000.0 then
s=format(d/1024.0/1024.0/1024.0,"0")+ " GigaBytes"
```

```

else
s=format(d/1024.0/1024.0,"0")+ " MegaBytes"
end if
else
s=format(d/1024.0,"0")+ " KiloBytes"
end if
else
s=format(d,"0")+ " Bytes"
end if

msgBox "Your drive with the system folder is "+s+" big."

```

Notes: This should be used to handle any volume size.
Return value changed in version 10.0 from double to SInt64.

Added Linux support in version 16.0.
Returns -1 on error.

Blog Entries

- [MBS Xojo Plugins, version 22.5pr5](#)
- [MBS Xojo Plugins, version 20.3pr1](#)
- [MBS Xojo Plugins, version 19.4pr1](#)
- [MBS Xojo / Real Studio Plugins, version 16.0pr7](#)
- [MBS Xojo / Real Studio Plugins, version 14.0pr1](#)

Xojo Developer Magazine

- [2.1, page 42: Details about disks, Getting the name, the size, and more information about a disc.](#) by Christian Schmitz

21.4.42 VolumeUUIDMBS as string

Plugin Version: 13.0, Platform: macOS, Targets: All.

Function: Queries volume UUID.

Notes: Requires Mac OS X 10.7 or newer.

Blog Entries

- [MBS Real Studio Plugins, version 13.0pr1](#)

21.4.43 WriteFileMBS(data as MemoryBlock) as boolean

Plugin Version: 19.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes a file.

Notes: Faster than `TextOutputStream` or `BinaryStream` on MacOS.

Returns true on success or false on failure.
See also `ReadFileMBS` function.

Limited to maximum size of a string (2 GB)
See also:

- 21.4.44 `WriteFileMBS(data as string)` as boolean

432

21.4.44 `WriteFileMBS(data as string)` as boolean

Plugin Version: 19.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes a file.

Notes: Faster than `TextOutputStream` or `BinaryStream` on MacOS.

Returns true on success or false on failure.
See also `ReadFileMBS` function.

Limited to maximum size of a string (2 GB)

Blog Entries

- [MBS Xojo Plugins in version 19.0](#)
- [MBS Xojo Plugins, version 18.6pr1](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- 21.1, page 28: [News from MBS Xojo Plugins, What's up with MonkeyBread Software by Stefanie Juchmes](#)
- 17.5, page 40: [What's New in the MBS Plugins, With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes](#)
- 17.2, page 10: [News](#)

See also:

- 21.4.43 `WriteFileMBS(data as MemoryBlock)` as boolean

431

21.4.45 Properties

21.4.46 AccessDateMBS(UTC as boolean = false) as date

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The access date of the file or folder.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var xd as date = f.AccessDateMBS
MsgBox xd.LongDate+" "+xd.LongTime
```

Notes: Setting this value is not supported on Linux currently.

To query Spotlight's Last Open day, please check the FAQ for sample code.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.47 AccessDateTimeMBS(UTC as boolean = false) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The access date of the file or folder.

Notes: Setting this value is not supported on Linux currently.

To query Spotlight's Last Open day, please check the FAQ for sample code.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.48 AddedToDirectoryDateMBS as date

Plugin Version: 16.4, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the date the file was added to the folder.

Notes: Requires OS X 10.10 or newer.

(Read and Write computed property)

21.4.49 AddedToDirectoryDateTimeMBS as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the date the file was added to the folder.

Notes: Requires OS X 10.10 or newer.

(Read and Write computed property)

21.4.50 `AttributeModificationDateMBS(UTC as boolean = false)` as date

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The attribute modification date of the file or folder.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var xd as date = f.AttributeModificationDateMBS
MsgBox xd.LongDate+" "+xd.LongTime
```

Notes: Attribute Modification dates are supported only on Mac OS X currently.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.51 `AttributeModificationDateTimeMBS(UTC as boolean = false)` as Date-Time

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The attribute modification date of the file or folder.

Notes: Attribute Modification dates are supported only on Mac OS X currently.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.52 `BackupDateMBS(UTC as boolean = false)` as date

Plugin Version: 10.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The backup date of the file or folder.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var xd as date = f.BackupDateMBS
MsgBox xd.LongDate+" "+xd.LongTime
```

Notes: Backup dates are supported only on Mac OS X currently.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.53 BackupDateTimeMBS(UTC as boolean = false) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The backup date of the file or folder.

Notes: Backup dates are supported only on Mac OS X currently.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.54 CreationDateMBS(UTC as boolean = false) as date

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The creation date of the file or folder.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var xd as date = f.CreationDateMBS
MsgBox xd.LongDate+" "+xd.LongTime
```

Notes: Setting this value is not supported on Linux currently.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.55 CreationDateTimeMBS(UTC as boolean = false) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The creation date of the file or folder.

Notes: Setting this value is not supported on Linux currently.

If UTC is true, you get/set date object where time is UTC.

(Read and Write computed property)

21.4.56 ModificationDateMBS(UTC as boolean = false) as date

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The modification date of the file or folder.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var xd as date = f.ModificationDateMBS
MsgBox xd.LongDate+" "+xd.LongTime
```

Notes: Setting this value is not supported on Linux currently.
If UTC is true, you get/set date object where time is UTC.
(Read and Write computed property)

21.4.57 ModificationDateTimeMBS(UTC as boolean = false) as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The modification date of the file or folder.

Notes: Setting this value is not supported on Linux currently.
If UTC is true, you get/set date object where time is UTC.
(Read and Write computed property)

21.5 class LargeBinaryStreamMBS

21.5.1 class LargeBinaryStreamMBS

Platforms: macOS, Linux, Windows, Targets: All.

Function: Allows you to access files with more than 2GB in size.

Notes: Deprecated in favor of BinaryStream class in Xojo unless you need a feature provided here, so please let us know!

Blog Entries

- [MBS Plugins updated for Xojo 2019r2](#)
- [Cleanup your C code for MacOS and remove FSCreateFileUnicode](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4](#)
- [MBS Xojo / Real Studio Plugins, version 16.4pr6](#)
- [MBS Xojo / Real Studio Plugins, version 16.4pr5](#)
- [MBS Real Studio Plugins, version 12.5pr2](#)
- [MBS Real Studio Plugins, version 12.4pr4](#)
- [MBS Real Studio Plugins, version 12.4pr3](#)

Videos

- [Presentation from London conference about MBS Plugins.](#)

21.5.2 Methods

21.5.3 Allocate(count as int64, flags as Integer) as int64

Plugin Version: 7.4, Platform: macOS, Targets: All.

Function: Allocates disk space for this file.

Example:

```
Var l as LargeBinaryStreamMBS // your stream  
Var d as int64
```

```
d=l.Allocate(1024*1024*1024,2) // 1 GB
```

Notes: Only for Mac OS.

The flags can be like this:

AllocDefaultFlags	0	as much as possible, not contiguous
AllocAllOrNothingMask	1	allocate all of the space, or nothing
AllocContiguousMask	2	new space must be one contiguous piece
AllocNoRoundUpMask	4	don't round up allocation to clump size

Note that the length property is not changed because the space is not used.

21.5.4 close

Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

Notes: There is no need to call this method except you want to free all resources of this object now without waiting for Xojo to do it for you.

21.5.5 Create(file as folderitem, MacType as string, MacCreator as string) as LargeBinaryStreamMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a file as a LargeBinaryStreamMBS.

Example:

```
Var f as FolderItem // your file
Var l as LargeBinaryStreamMBS
```

```
l = LargeBinaryStreamMBS.Create(f, "TEXT", "ttx")
```

Notes: If there is already a file, it is deleted.

On Windows the parameters are ignored.

Returns nil on any error.

See also:

- 21.5.6 Create(path as string, MacType as string, MacCreator as string, WinShareMode as Integer = 0) as LargeBinaryStreamMBS 438

21.5.6 Create(path as string, MacType as string, MacCreator as string, WinShareMode as Integer = 0) as LargeBinaryStreamMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a file as a LargeBinaryStreamMBS.

Example:

```
Var l as LargeBinaryStreamMBS
l = LargeBinaryStreamMBS.Create("C:\test.txt", "", "")
```

Notes: If there is already a file, it is deleted.

On Windows the parameters are ignored.

Returns nil on any error.

See also:

- 21.5.5 Create(file as folderitem, MacType as string, MacCreator as string) as LargeBinaryStreamMBS
438

21.5.7 CreateResStream(file as folderitem, MacType as string, MacCreator as string) as ResStreamMBS

Plugin Version: 12.4, Platform: macOS, Targets: All.

Function: Creates a new ResStreamMBS.

Notes: If there is already a file, it is deleted.

If the file could not be created it is deleted.

Returns nil on any error.

See also:

- 21.5.8 CreateResStream(path as string, MacType as string, MacCreator as string) as ResStreamMBS
439

21.5.8 CreateResStream(path as string, MacType as string, MacCreator as string) as ResStreamMBS

Plugin Version: 12.4, Platform: macOS, Targets: All.

Function: Creates a new ResStreamMBS.

Notes: If there is already a file, it is deleted.

If the file could not be created it is deleted.

Returns nil on any error.

See also:

- 21.5.7 CreateResStream(file as folderitem, MacType as string, MacCreator as string) as ResStreamMBS
439

21.5.9 DeleteDataFork(file as folderitem)

Plugin Version: 12.4, Platform: macOS, Targets: All.

Function: Deletes the data fork of a file.

Notes: Equal to open the file using a `BinaryStream` and setting the `length` property to 0.

On Mac OS a file can exist without a datafork, with a resource fork or even without any fork.

21.5.10 DeleteResourceFork(file as folderitem)

Plugin Version: 12.4, Platform: macOS, Targets: All.

Function: Deletes the resource fork of a file.

Notes: Equal to open the file using a `ResStreamMBS` and setting the `length` property to 0.

On Mac OS a file can exist without a datafork, with a resource fork or even without any fork.

21.5.11 Flush

Plugin Version: 5.0, Platforms: macOS, Linux, Targets: All.

Function: Makes sure every bit of the stream is written to disc.

Notes: Only for Mac OS X.

21.5.12 LockFileExclusive as boolean

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Locks the file so no other app can access it.

Notes: Returns true on success.

It may be possible that there is still a way around the lock.

21.5.13 Open(file as folderitem, write as Boolean) as LargeBinaryStreamMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens a file as a `LargeBinaryStreamMBS`.

Example:

```
Var l as LargeBinaryStreamMBS
```

```
Var f as FolderItem = SpecialFolder.Desktop.Child("testfile")
```

```
l = LargeBinaryStreamMBS.Open(f, true)
```

Notes: Returns nil on any error.

See also:

- 21.5.14 `Open(path as string, write as Boolean, WinShareMode as Integer = 0)` as `LargeBinaryStreamMBS` 441

21.5.14 `Open(path as string, write as Boolean, WinShareMode as Integer = 0)` as `LargeBinaryStreamMBS`

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens a file as a `LargeBinaryStreamMBS`.

Example:

```
// write to parallel port:
Var b as LargeBinaryStreamMBS = LargeBinaryStreamMBS.Open("LPT1", true, 3)
b.Write "Hello World"
b.Close
```

Notes: Returns nil on any error.

For special cases, you may need to allow Shared Read or Write and for that case, we have `WinShareMode` parameter.

For `WinShareMode`, you can pass 1 for shared reading, 2 for shared writing, 4 for shared deletion. Or combine those. Passing zero prevents sharing.

See also:

- 21.5.13 `Open(file as folderitem, write as Boolean)` as `LargeBinaryStreamMBS` 440

21.5.15 `OpenAsResStream(file as folderitem, write as Boolean)` as `ResStreamMBS`

Plugin Version: 12.4, Platform: macOS, Targets: All.

Function: Opens a file's resourcefork as a `ResStreamMBS`.

Notes: Returns nil on any error.

See also:

- 21.5.16 `OpenAsResStream(path as string, write as Boolean)` as `ResStreamMBS` 441

21.5.16 `OpenAsResStream(path as string, write as Boolean)` as `ResStreamMBS`

Plugin Version: 12.4, Platform: macOS, Targets: All.

Function: Opens a file's resourcefork as a `ResStreamMBS`.

Notes: Returns nil on any error.

See also:

- 21.5.15 `OpenAsResStream(file as folderitem, write as Boolean)` as `ResStreamMBS` 441

21.5.17 QueryDiskGeometry(byref Cylinders as Int64, byref MediaType as Integer, byref TracksPerCylinder as Integer, byref SectorsPerTrack as Integer, byref BytesPerSector as Integer) as boolean

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries disk size on Windows.

Notes: This function is only useful on Windows and only if you opened a physical disc.
Returns true on success.

21.5.18 Read(count as Integer) as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes into a string.

Example:

```
Var b as LargeBinaryStreamMBS // your stream  
Var s as string
```

```
s=b.read(5)
```

21.5.19 ReadBlock(count as Integer) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes into a memoryblock.

Example:

```
Var b as LargeBinaryStreamMBS // your stream  
Var s as memoryblock
```

```
s=b.read(5)
```

21.5.20 Readbyte as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads an 8bit Byte from the stream.

Example:

```
Var b as LargeBinaryStreamMBS // your stream  
Var i as Integer
```

```
i=B.readbyte
```

21.5.21 ReadLong as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a signed 32bit Integer from the stream.

Example:

```
Var b as LargeBinaryStreamMBS // your stream  
Var i as Integer
```

```
i=B.readlong
```

Notes: This function is affected by the LittleEndian Setting.

21.5.22 ReadShort as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a signed 16bit Integer from the stream.

Example:

```
Var b as LargeBinaryStreamMBS // your stream  
Var i as Integer
```

```
i=B.readshort
```

Notes: This function is affected by the LittleEndian Setting.

21.5.23 UnlockFileExclusive as boolean

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Unlocks the file so other applications can possibly access it.

Notes: Returns true on success.

21.5.24 WinCreateStream(file as folderitem, StreamName as String, WinShareMode as Integer = 0) as LargeBinaryStreamMBS

Plugin Version: 16.4, Platform: Windows, Targets: All.

Function: Creates a named stream in a file.

Example:

```

Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var l as LargeBinaryStreamMBS = LargeBinaryStreamMBS.WinCreateStream(f, "test")
if l = nil then
  MsgBox "Error"
else
  l.Write "Hello"
  l.close
end if

```

Notes: Returns nil in case of error.

21.5.25 WinDeleteStream(file as folderitem, StreamName as String) as boolean

Plugin Version: 16.4, Platform: Windows, Targets: All.

Function: Deletes a named stream in a file.

Example:

```

Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
if LargeBinaryStreamMBS.WinDeleteStream(f) then
  MsgBox "OK"
else
  MsgBox "Error"
end if

```

Notes: Returns true on success.

21.5.26 WinOpenStream(file as folderitem, StreamName as String, write as Boolean, WinShareMode as Integer = 0) as LargeBinaryStreamMBS

Plugin Version: 16.4, Platform: Windows, Targets: All.

Function: Opens a named stream in a file.

Example:

```

Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var l as LargeBinaryStreamMBS = LargeBinaryStreamMBS.WinOpenStream(f, "test", false)

```

```
if l = nil then
  MsgBox "Error"
else
  MsgBox l.Read(l.Length)
end if
```

Notes: Returns nil in case of error.

21.5.27 Write(data as string)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes bytes from a string to file.

21.5.28 WriteBlock(data as memoryblock, count as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes count bytes from a memoryblock to file.

Example:

```
Var b as LargeBinaryStreamMBS // your stream
Var m as memoryblock
```

```
b.writeblock m, m.size
```

21.5.29 WriteByte(data as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes a byte to file.

21.5.30 WriteLong(data as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes an 32bit integer to file.

Notes: This method is affected by the LittleEndian Setting.

21.5.31 WriteShort(data as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes an 16bit integer to file.

Notes: This method is affected by the LittleEndian Setting.

21.5.32 Properties

21.5.33 CanWrite as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if you are allowed to write.

Notes: Even if you open a file for write you may still not be allowed to write if the permissions of the file don't allow you to write.

(Read only property)

21.5.34 EOF as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if you are at the end of the stream.

Notes: You can set this property to truncate the file.

(Read only property)

21.5.35 Lasterror as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error code.

Notes: The values are platform dependent, but zero is no error everywhere.

(Read and Write property)

21.5.36 Length as Int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the current length of the file.

Notes: You can truncate the stream by setting this property.

Can't be set on Linux currently.

(Read and Write property)

21.5.37 LittleEndian as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies if Integers and Shorts shall be converted in their endianness when read or written.

Notes: See Xojo's `binarystream` for more details.

For native platform you may set `"littleendian=targetwin32"`.

(Read and Write property)

21.5.38 Position as Int64

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns current position.

Notes: You can set the current file stream position using this property.

(Read and Write property)

21.5.39 Yield as Boolean

Plugin Version: 9.5, Platform: macOS, Targets: All.

Function: Whether time should be given to other threads.

Notes: If true on Mac OS X CPU time will be given to other threads while read or write operations are pending. To have an effect you need to call read/write methods in a thread.

(Read and Write property)

21.6 class ResStreamMBS

21.6.1 class ResStreamMBS

Platform: macOS, Targets: All.

Function: Allows you to access a resourcefork in its binary representation.

Notes: Useful to copy the resourcefork faster from one file to another.

Subclass of the LargeBinaryStreamMBS class.

Blog Entries

- [MBS Plugins updated for Xojo 2019r2](#)

21.7 class StdinMBS

21.7.1 class StdinMBS

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: A class to read from stdin.

Example:

```
Var s as new StdoutMBS
```

```
Var r as new StdinMBS
```

```
Var v,n as Integer
```

```
s.Write "Integer: "
```

```
n=r.ReadInteger(v)
```

```
print "integer read: "+str(v)
```

```
print "integer count: "+str(n)
```

```
Var d as Double
```

```
s.Write "Double: "
```

```
n=r.ReadDouble(d)
```

```
print "double read: "+str(d)
```

```
print "double count: "+str(n)
```

```
Var t as string
```

```
s.Write "String: "
```

```
n=r.ReadString(t)
```

```
print "string read: "+t
```

```
print "string count: "+str(n)
```

```
do
```

```
s.Write "Character (type a and return to end): "
```

```
n=r.GetCharacter
```

```
print "got character: "+str(n)
```

```
loop until n=65 or n=97
```

Blog Entries

- [MBS Xojo Plugins, version 20.5pr2](#)
- [Console and GUI in one project](#)

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.4](#)
- [MBS Xojo Plugins, version 17.4pr6](#)
- [MBS Real Studio Plugins, version 12.5pr8](#)
- [MBS Real Studio Plugins, version 12.3pr4](#)

21.7.2 Methods

21.7.3 AttachConsole(ProcessID as Integer = -1) as Integer

Plugin Version: 12.5, Platform: Windows, Targets: Desktop, Console & Web.

Function: Attaches the calling process to the console of the specified process.

Notes: ProcessId: The identifier of the process whose console is to be used. This parameter can be one of the following values.

Value	Meaning
pid	Use the console of the specified process.
-1	Use the console of the parent of the current process.

Returns Windows error code or zero for success.

A process can be attached to at most one console. If the calling process is already attached to a console, the error code returned is `ERROR_ACCESS_DENIED` (5). If the specified process does not have a console, the error code returned is `ERROR_INVALID_HANDLE` (6). If the specified process does not exist, the error code returned is `ERROR_GEN_FAILURE` (31).

21.7.4 Flush

Plugin Version: 12.3, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Flushes input.

21.7.5 FreeConsole as Integer

Plugin Version: 12.5, Platform: Windows, Targets: Desktop, Console & Web.

Function: Detaches the calling process from its console.

Notes: Returns Windows error code or zero for success.

A process can be attached to at most one console. If the calling process is not already attached to a console, the error code returned is `ERROR_INVALID_PARAMETER` (87).

A process can use the `FreeConsole` function to detach itself from its console. If other processes share the console, the console is not destroyed, but the process that called `FreeConsole` cannot refer to it. A console is closed when the last process attached to it terminates or calls `FreeConsole`. After a process calls `FreeConsole`, it can call the `AllocConsole` function to create a new console or `AttachConsole` to attach to another console.

21.7.6 GetCharacter as Integer

Plugin Version: 8.7, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Reads in one character and returns the ASCII code.

Notes: Returns one if a value was read and 0 if not.

21.7.7 Read(count as Integer) as string

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reads data from stdin.

Notes: Data is returned in binary encoding.

21.7.8 ReadDouble(byref value as Double) as Integer

Plugin Version: 8.7, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reads a double from the console.

Notes: Returns one if a value was read and 0 if not.

21.7.9 ReadInteger(byref value as Integer) as Integer

Plugin Version: 8.7, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reads an integer from the console.

Notes: Returns one if a value was read and 0 if not.

21.7.10 ReadString(byref value as string) as Integer

Plugin Version: 8.7, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reads a string from the console.

Notes: This string is limited to 4095 characters.

Returns one if a value was read and 0 if not.

21.7.11 Properties

21.7.12 Echo as Boolean

Plugin Version: 17.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Enables/disables echo for stdin.

Example:

```
// try this on Mac/Linux GUI/Console app with the app launched from Terminal,
// or in console app on Windows launched from command prompt
```

```
Var s as new StdoutMBS
```

```
s.Write "Hello World"+chr(10)
```

```
Var t as string
```

```
Var n as integer = StdinMBS.ReadString(t)
```

```
s.Write chr(10)
```

```
s.Write "Read with echo: "+t+chr(10)
```

```
StdinMBS.Echo = false
```

```
n = StdinMBS.ReadString(t)
```

```
s.Write chr(10)
```

```
s.Write "Read without echo: "+t+chr(10)
```

```
StdinMBS.Echo = true
```

Notes: You can disable echo for password input.

By default it is enabled.

(Read and Write property)

21.7.13 IsReady as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Checks whether there is data waiting in input buffer to be read.

Example:

```
Var d As New date
```

```
Do
// print time if second changed
Var e As New date
If e.Second <>d.Second Then
Print e.LongTime
d = e
End If

// if ready, read a key
If StdinMBS.IsReady Then
Var s As String
Var l As Integer = StdinMBS.ReadString(s)
Print "Input "+Str(l)+"": "+s
End If
Loop
```

Notes: If `IsReady` returns false and you would call `ReadString` method, the app could be blocked until user types something on console.

Added Windows support in version 20.5. We return true, if there is a return in the input buffer, so you can call `Read()` functions.

(Read only property)

21.8 class StdoutMBS

21.8.1 class StdoutMBS

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: A class to write to stdout.

Example:

```
Var s as new StdoutMBS
```

```
Var r as new StdinMBS
```

```
Var v,n as Integer
```

```
s.Write "Integer: "
```

```
n=r.ReadInteger(v)
```

```
print "integer read: "+str(v)
```

```
print "integer count: "+str(n)
```

```
Var d as Double
```

```
s.Write "Double: "
```

```
n=r.ReadDouble(d)
```

```
print "double read: "+str(d)
```

```
print "double count: "+str(n)
```

```
Var t as string
```

```
s.Write "String: "
```

```
n=r.ReadString(t)
```

```
print "string read: "+t
```

```
print "string count: "+str(n)
```

```
do
```

```
s.Write "Character (type a and return to end): "
```

```
n=r.GetCharacter
```

```
print "got character: "+str(n)
```

```
loop until n=65 or n=97
```

Blog Entries

- [Console and GUI in one project](#)
- [Tip of the day: Console support in GUI app](#)

- [MBS Real Studio Plugins, version 12.5pr8](#)
- [Tip of the day: Mac Shell Scripting](#)

21.8.2 Methods

21.8.3 AttachConsole(ProcessID as Integer = -1) as Integer

Plugin Version: 12.5, Platform: Windows, Targets: Desktop, Console & Web.

Function: Attaches the calling process to the console of the specified process.

Notes: ProcessId: The identifier of the process whose console is to be used. This parameter can be one of the following values.

Value	Meaning
pid	Use the console of the specified process.
-1	Use the console of the parent of the current process.

Returns Windows error code or zero for success.

A process can be attached to at most one console. If the calling process is already attached to a console, the error code returned is `ERROR_ACCESS_DENIED` (5). If the specified process does not have a console, the error code returned is `ERROR_INVALID_HANDLE` (6). If the specified process does not exist, the error code returned is `ERROR_GEN_FAILURE` (31).

21.8.4 Flush

Plugin Version: 8.7, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Makes sure that all data written using `Write` is already on the console.

21.8.5 FreeConsole as Integer

Plugin Version: 12.5, Platform: Windows, Targets: Desktop, Console & Web.

Function: Detaches the calling process from its console.

Notes: Returns Windows error code or zero for success.

A process can be attached to at most one console. If the calling process is not already attached to a console, the error code returned is `ERROR_INVALID_PARAMETER` (87).

A process can use the `FreeConsole` function to detach itself from its console. If other processes share the console, the console is not destroyed, but the process that called `FreeConsole` cannot refer to it. A console is closed when the last process attached to it terminates or calls `FreeConsole`. After a process calls `FreeConsole`, it can call the `AllocConsole` function to create a new console or `AttachConsole` to attach to another console.

21.8.6 Write(data as string)

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Writes data to stdout.

Notes: You have to make sure your data is in good text encoding.

Chapter 22

Fonts

Chapter 23

Graphics & Pictures

23.1 class Graphics

23.1.1 class Graphics

Platforms: macOS, Linux, Windows, Targets: All.

Function: Extends Xojo's Graphics Class.

23.1.2 Methods

23.1.3 DrawRotatedTextMBS(Rotation as Double, text as string, x as Integer, y as Integer, Center as Boolean = false, alpha as Double = 1.0, NoSwapY as boolean = false, FontWidth as Integer = 0)

Plugin Version: 11.2, Platforms: macOS, Windows, Targets: Desktop only.

Function: Draws rotated text.

Example:

```
Sub Paint(g As Graphics)
```

```
g.DrawRotatedTextMBS 45, "Hello World", 100, 100
```

```
End Sub
```

Notes: Rotation is the angle in degree.

Currently only supported for Mac OS X and Windows in GUI applications.

The plugin does not see the clipping offset for the graphics object. So for graphics objects from printer or canvas, you need to offset the position. Seems like with a Canvas the coordinates are relative to canvas in Cocoa, but relative to Windows in Carbon.

We have the optional parameter `Center` which tells the plugin to center the rotated text, so the rotation point is in the middle of the text.

The plugin queries `Bold`, `Italic`, `Underline`, `ForeColor`, `TextUnit`, `TextSize` or `TextFont`.

`Alpha` gives the alpha value for Mac OS X. Can be 1.0 (Full color) to 0.0 (no color).

On Mac targets the Y coordinate needs to be swapped internally from top down coordinates to bottom up coordinates. As this does not work 100% correct on printers (the plugin doesn't know page margins), you can provide correct Y coordinate yourself and pass `NoSwapY = true`. (added in plugin version 13.1)

On Windows the OS function used (`ExtTextOut`) does not support multiple lines.

`FontWidth` is only for Windows to define the width of the font. Default is 0 to use default width.

The average width, in logical units, of characters in the font. If `FontWidth` is zero, the aspect ratio of the device is matched against the digitization aspect ratio of the available fonts to find the

On Windows for drawing in a picture, make sure the font size and font name are set. Else it may not draw. For Windows, we only use HDC for drawing, so if you have a graphics context not using a HDC, this function doesn't draw. The function is not updated for `DirectDraw`.

Blog Entries

- [MBS Xojo Plugins, version 24.4pr1](#)
- [MBS Xojo Plugins, version 21.2pr1](#)
- [MBS Xojo Plugins, version 18.2pr1](#)
- [MonkeyBread Software Releases the MBS Xojo plug-ins in version 17.0](#)
- [MBS Xojo / Real Studio Plugins, version 17.0pr4](#)
- [MBS Real Studio Plugins, version 13.1pr1](#)
- [MBS Real Studio Plugins, version 11.4pr2](#)
- [MBS Real Studio Plugins, version 11.3fc](#)
- [MBS Real Studio Plugins, version 11.3pr8](#)
- [Rotated text](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [17.5, page 42: What's New in the MBS Plugins, With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes](#)
- [15.2, page 10: News](#)
- [10.1, page 75: Using Plugins, Working with the Monkeybread Plugins by Marc Zeedar](#)

23.1.4 MeasureRotatedTextMBS(text as string, byref Width as Double, byref Height as Double, FontWidth as Integer = 0) as Boolean

Plugin Version: 17.0, Platforms: macOS, Windows, Targets: Desktop only.

Function: Measure size of text.

Notes: This is compatible to text drawing in DrawRotatedTextMBS, so you can use it to calculate the required space.

No rotation is used for measurement.

Returns true on success and false on failure.

Blog Entries

- [MBS Xojo Plugins, version 21.2pr1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 18.1](#)
- [MBS Xojo Plugins, version 18.1pr1](#)
- [MonkeyBread Software Releases the MBS Xojo plug-ins in version 17.0](#)
- [MBS Xojo / Real Studio Plugins, version 17.0pr3](#)

Xojo Developer Magazine

- [15.2, page 10: News](#)

Chapter 24

Hotkey

24.1 class HotKeyMBS

24.1.1 class HotKeyMBS

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The class for registering hotkeys.

Notes: This is a crossplatform replacement for our CarbonHotKeyMBS class.

Blog Entries

- [MBS Xojo Plugins, version 24.3pr1](#)
- [MBS Xojo Plugins, version 22.2pr1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.4](#)
- [MBS Xojo Plugins, version 20.4pr6](#)
- [MBS Xojo Plugins, version 18.4pr2](#)
- [MBS Xojo / Real Studio Plugins, version 15.3pr1](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 15.2](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr6](#)

Xojo Developer Magazine

- [18.6, page 10: News](#)
- [13.5, page 8: News](#)

24.1.2 Methods

24.1.3 Close

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Closes hotkey.

Notes: Same as destructor, but to unregister hotkey now.

24.1.4 Constructor(KeyCode as Integer, Modifiers as Integer, Exclusive as Boolean = false)

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Registers a new keycode.

Notes: The plugin may not check if combination is allowed, makes sense or can be pressed.

24.1.5 KeyCodeForText(name as string) as Integer

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Queries a keycode for a character.

Example:

```
MsgBox "Return: " + str(HotKeyMBS.KeyCodeForText("return"))
```

Notes: Helper function to return platform specific key codes for various keys.

Plugin has a list based on the constants defined by Apple and Microsoft for keys.

24.1.6 Properties

24.1.7 Exclusive as Boolean

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Whether exclusive ownership of hotkey was requested.

Notes: (Read only property)

24.1.8 Handle as Integer

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The internal object reference.

Notes: Only used on Mac OS X.

(Read only property)

24.1.9 ID as Integer

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The internal ID of the hotkey.

Notes: (Read only property)

24.1.10 KeyCode as Integer

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The keycode registered.

Notes: (Read only property)

24.1.11 Modifiers as Integer

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The modifiers used at registration.

Notes: (Read only property)

24.1.12 Events

24.1.13 KeyDown

Plugin Version: 15.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: The key down event.

24.1.14 KeyUp

Plugin Version: 15.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: The key up event.

24.1.15 Constants

Modifier Flags

Constant	Value	Description
AlphaKey	&h400	Alpha Lock
CommandKey	&h100	Command Key on Mac, Windows Key on Windows.
ControlKey	&h1000	Control Key
OptionKey	&h800	Option/Alt Key
ShiftKey	&h200	Shift Key

Chapter 25

JavaScript Object Notation

25.1 class JSONEntryMBS

25.1.1 class JSONEntryMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for an entry in a JSON object or array.

Example:

```
Var o As New JSONMBS
```

```
o.add 1  
o.add 2  
o.add 3
```

```
For Each v As JSONEntryMBS In o.IterateEntries  
Break  
// watch in debugger  
Next
```

Notes: For objects, we use the keys as names here and for arrays the index.

Blog Entries

- [News from the MBS Xojo Plugins Version 23.5](#)
- [Moving to new JSONMBS class](#)
- [New JSON Plugin for Xojo](#)

25.1.2 Methods

25.1.3 Constructor

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The private constructor.

See also:

- 25.1.4 Constructor(other as JSONEntryMBS) 468

25.1.4 Constructor(other as JSONEntryMBS)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The copy constructor.

See also:

- 25.1.3 Constructor 468

25.1.5 Properties

25.1.6 Item as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The value of the entry as variant.

Notes: (Read only property)

25.1.7 Name as String

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The key name in the JSON object.

Notes: (Read only property)

25.2 class JSONIteratorMBS

25.2.1 class JSONIteratorMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The iterator object use for for-each loops in Xojo.

Notes: Implements Iterable and Iterator interfaces.

This class is hidden for auto complete as you don't need to use it manually. Please use it using for each loops.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [News from the MBS Xojo Plugins Version 23.5](#)
- [Embracing for each loops](#)

Interfaces: Iterator, Iterable

25.2.2 Methods

25.2.3 Constructor

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The private constructor.

25.2.4 Iterator as Iterator

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries the iterator object.

Example:

```
Var o As New JSONMBS
```

```
o.add 1  
o.add 2  
o.add 3
```

```
For Each v As JSONMBS In o.Iterate  
Break
```

```
// watch in debugger  
Next
```

Notes: Returns self and is used internally by Xojo for the for each loops.
Part of the Iterable interface and needed for for-each loops.

25.2.5 MoveNext as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Moves to next object for iterator.

Notes: Returns true on success and false in case of an error like end of iterator.
Part of the Iterator interface and needed for for-each loops.

25.2.6 Value as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Current value for the iterator.

Notes: For an iterator over objects, this is a JSONMBS object.

For an iterator over entries, this is a JSONEntryMBS object.

For an iterator over values, this is a variant with the value.

Part of the Iterator interface and needed for for-each loops.

25.2.7 Properties

25.2.8 Index as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Current index in array or object.

Notes: We iterate over the items in an array or object.

Going by index avoids crashing if you edit the object/array, but going by index may skip an item if you added one before the current one or skip one if you remove one.

(Read only property)

25.2.9 Root as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The root object reference.

Notes: (Read only property)

25.3 class JSONMBS

25.3.1 class JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for JSON object parsing and creating.

Example:

```
Var o as JSONMBS = JSONMBS.NewObjectNode
```

```
o.AddItemToObject "Hello", JSONMBS.NewNumberNode(1)
```

```
o.AddItemToObject "World", JSONMBS.NewNumberNode(2)
```

```
MsgBox o.toString
```

Notes: Our JSON class

- has a high performance.
- is based on a modern C++ library
- supports Xojo features like for-each loops with iterators
- is almost compatible to the old JSONMBS class from before
- is almost compatible to JSONItem class
- automatically converts between variant and JSONMBS as needed.
- reports errors with JSONException class.
- can query, search and replace JSON values with JSON Paths.
- can flatten and unflatten JSON.
- can create patches and merge them back with two ways.
- provides a lot of convenience functions.
- can convert from JSON to/from dictionaries and arrays.
- handles currency better and preserves digits.
- has a lot of duplicate method/property names for API 1 and 2 compatibility and you can choose whether you prefer Append vs Add, Insert vs. AddAt or Key vs. NameAt.
- preserves the order of object entries and allows explicit sorting.
- can convert to HTML for displaying itself.
- prefers case sensitive, but can optionally do case insensitive.

We can convert integer, boolean, string, double, arrays, dictionaries to JSON internally. And convert back to variant containing integer, boolean, string, double, arrays and dictionaries.

We back ported things like GetChildNode, GetNextNode and GetPreviousNode, but those are not very efficient and you may want to move to using iterators, query values/entries/keys arrays or access items by index.

The old class is still available in a separated OldJSON Plugin for our customers for compatibility with class renamed to OldJSONMBS.

Blog Entries

- [News from the MBS Xojo Plugins Version 24.2](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.5](#)
- [New JSON Plugin for Xojo](#)
- [News from the MBS Xojo Plugins in Version 23.0](#)
- [News from the MBS Xojo Plugins Version 22.5](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [Extends MBS Plugin classes in Xojo](#)
- [JSON Sort](#)
- [MBS Xojo Plugins in version 19.2](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.1](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [22.6, page 9: News](#)
- [22.4, page 9: News](#)
- [21.5, page 10: News](#)
- [21.1, pages 95 to 96: A Database That Connects, A Mongo DB photo database used with FileMaker and Xojo by Stefanie Juchmes](#)
- [19.3, page 10: News](#)
- [18.6, page 10: News](#)
- [17.5, page 9: News](#)
- [17.4, page 10: News](#)
- [16.4, page 11: News](#)
- [15.2, page 10: News](#)

25.3.2 Methods

25.3.3 Add(Value as Variant)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a value to an array.

Example:

```
Var o As New JSONMBS
```

```
o.Add 1  
o.Add "Hello"  
o.Add JSONMBS.NewNullNode
```

```
MessageBox o.toString
```

Notes: Variant is converted to JSONMBS if needed.

If the self is an empty object, we replace it with an empty array and add the value.

Same as Add method.

25.3.4 AddAt(index As Integer, value As Variant)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Insert a value to an array at the given index.

Example:

```
Var o As New JSONMBS
```

```
o.Append 1  
o.AddAt 1,2
```

```
MessageBox o.toString
```

```
o.AddAt 1,3
```

```
MessageBox o.toString
```

Notes: Variant is converted to JSONMBS if needed.

If the self is an empty object, we replace it with an empty array and insert the value.

Same as Insert method.

25.3.5 AddItemToArray(item as JSONMBS)

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an item to an array.

Example:

```

Var subscriptionId As Integer = 123456
Var Quantity As Integer = 1

// build inner object
Var ji As JSONMBS = JSONMBS.NewObjectNode

ji.AddItemToObject "subscriptionId", JSONMBS.NewNumberNode(subscriptionId)
ji.AddItemToObject "newQuantity", JSONMBS.NewNumberNode(Quantity)

// array around it
Var ja As JSONMBS = JSONMBS.NewArrayNode
ja.AddItemToArray ji

// and object around
Var jo As JSONMBS = JSONMBS.NewObjectNode
jo.AddItemToObject "subscriptionAmendmentParameters", ja

Var json As String = jo.toString
MessageBox json

' {
' "subscriptionAmendmentParameters": [
' {
' "subscriptionId": 123456,
' "newQuantity": 1
' }
' ]
' }

```

Notes: If you add item from new nodes created with plugin, we add them to the tree. If you add items from existing node from other JSON tree, we add references.

25.3.6 AddItemToObject(label as string, value as JSONMBS)

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an item to an object with given label.

Example:

```

Var subscriptionId As Integer = 123456
Var Quantity As Integer = 1

// build inner object
Var ji As JSONMBS = JSONMBS.NewObjectNode

ji.AddItemToObject "subscriptionId", JSONMBS.NewNumberNode(subscriptionId)
ji.AddItemToObject "newQuantity", JSONMBS.NewNumberNode(Quantity)

// array around it
Var ja As JSONMBS = JSONMBS.NewArrayNode
ja.AddItemToArray ji

// and object around
Var jo As JSONMBS = JSONMBS.NewObjectNode
jo.AddItemToObject "subscriptionAmendmentParameters", ja

Var json As String = jo.toString
MessageBox json

' {
' "subscriptionAmendmentParameters": [
' {
' "subscriptionId": 123456,
' "newQuantity": 1
' }
' ]
' }

```

Notes: If you add item from new nodes created with plugin, we add them to the tree.

If you add items from existing node from other JSON tree, we add references.

25.3.7 AddOrReplaceItemToObject(label as string, value as JSONMBS)

Plugin Version: 22.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an item to an object with given label.

Notes: Same as AddItemToObject, but removes old item for the label if it exists.

25.3.8 Append(Value as Variant)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Appends a value to an array.

Example:

Var o As New JSONMBS

```
o.Append 1
o.Append "Hello"
o.Append JSONMBS.NewNullNode
```

MessageBox o.toString

Notes: Variant is converted to JSONMBS if needed.

If the self is an empty object, we replace it with an empty array and add the value.

Same as Add method.

25.3.9 ApplyMergePatch(target as JSONMBS, patch as JSONMBS) as JSONMBS

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies a merge patch to a json document.

Example:

```
Var source As New JSONMBS(" { "+_
"  ""title"": ""Goodbye!"" ,"+_
"  ""author"": { "+_
"    ""givenName"": ""John"" ,"+_
"    ""familyName"": ""Doe"""+_
"  } ,"+_
"  ""tags"": [ ""example"", ""sample"" ] ,"+_
"  ""content"": ""This will be unchanged"""+_
" } ")
```

```
Var patch As New JSONMBS(" { "+_
"  ""title"": ""Hello!"" ,"+_
"  ""phoneNumber"": ""+01-123-456-7890"" ,"+_
```

```

" ""author"": { "+_
" ""familyName"": null"+_
" } ,"+_
" ""tags"": [ ""example"" ] "+_
" } ")

```

```
Var result As JSONMBS = JSONMBS.ApplyMergePatch(source, patch)
```

```
Var sourceText As String = source.toString(True)
```

```
Var resultText As String = result.toString(True)
```

```
Var PatchText As String = Patch.toString(True)
```

Break

Notes: The mergepatch function implement the IETF standard JSON Merge Patch

25.3.10 ApplyPatch(target as JSONMBS, patch as JSONMBS) as JSONMBS

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies a patch to a json document.

Example:

```

Var source As New JSONMBS(" { ""foo"": ""bar"" } ")
Var patch As New JSONMBS(" "+_
" { ""op"": ""add"", ""path"": ""/baz"", ""value"": ""qux"" } ,"+_
" { ""op"": ""add"", ""path"": ""/foo"", ""value"": [ ""bar"", ""baz"" ] } "+_
" ] ")

```

```
Var result As JSONMBS = JSONMBS.ApplyPatch(source, patch)
```

```
Var sourceText As String = source.toString(True)
```

```
Var resultText As String = result.toString(True)
```

```
Var PatchText As String = Patch.toString(True)
```

Break

Notes: The jsonpatch functions implement the IETF standard JavaScript Object Notation (JSON) Patch.

The JSON Patch IETF standard requires that the JSON Patch method is atomic, so that if any JSON Patch operation results in an error, the target document is unchanged. The patch function implements this requirement by generating the inverse commands and building an undo stack, which is executed if any part of the patch fails.

25.3.11 ArrayItem(index as integer, Clone as Boolean = false) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries array item with given index.

Notes: Added Clone parameter for version 20.4.

If Clone is true, we duplicate the JSON, so the new JSONMBS object does not point to origin JSONMBS object and exist independent.

25.3.12 ArrayItems(Clone as Boolean = false) as JSONMBS()

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries array items.

Notes: If Clone is true, we duplicate the JSON, so the new JSONMBS objects do not point to origin JSONMBS object and exist independent.

25.3.13 Clear

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears the JSON object.

Example:

```
Var j As New JSONMBS
```

```
j.add 1  
j.add 2  
j.Clear // remove all  
j.add 3
```

```
MessageBox j.toString(false)
```

Notes: Removes all entries from array or object.

25.3.14 Clone as JSONMBS

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clones the JSON tree.

25.3.15 Close

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Frees node.

Notes: No need to call this as destructor does the same.
But this call allows you to release circular references.

25.3.16 Compare(Other as JSONMBS) as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares two objects and returns

Example:

```
Var o As New JSONMBS
```

```
o.add 1
o.add 2
```

```
Var j As New JSONMBS
```

```
j.add 1
j.add 2
```

```
If j.Compare(o) = 0 Then
  MsgBox "equal"
Else
  MsgBox "not equal"
End If
```

Notes: Returns 0 if both are equal, 1 if bigger and -1 if smaller.

25.3.17 Constructor

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an empty object.

Example:

```
Var c As Currency = 1.2345
Var d as Double = c
```

```
Var j As New JSONItem
j.Value("test") = d // can't use c here!
```



```
Var m As New JSONMBS
m.Value("test") = c
```

```
MessageBox j.ToString+EndOfLine+m.toString
// shows { "test":1.234499999999999307 } vs { "test":1.2345 } since we store currency exactly.
```

See also:

- 25.3.18 Constructor(dic As Dictionary) 481
- 25.3.19 Constructor(JSONString as String) 481
- 25.3.20 Constructor(other as JSONMBS) 482

25.3.18 Constructor(dic As Dictionary)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new object with the content of the dictionary.

Example:

```
Var d As New Dictionary
d.Value(1) = 2
d.Value("Hello") = "World"
```

```
Var j As New JSONMBS(d)
```

```
MessageBox j.toString(False)
```

Notes: Converts all dictionary values to JSON objects internally.

See also:

- 25.3.17 Constructor 480
- 25.3.19 Constructor(JSONString as String) 481
- 25.3.20 Constructor(other as JSONMBS) 482

25.3.19 Constructor(JSONString as String)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Parses the given JSON String into the tree where this node is the root.

Example:

```
Var o as new JSONMBS(" { ""text"":""Hello World"" } ")
MsgBox o.toString
```

Notes: Text should be UTF-8.
 Raises an exception in case of error.
 See also:

- 25.3.17 Constructor 480
- 25.3.18 Constructor(dic As Dictionary) 481
- 25.3.20 Constructor(other as JSONMBS) 482

25.3.20 Constructor(other as JSONMBS)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copy constructor.

Notes: Makes a copy of all the nodes.

See also:

- 25.3.17 Constructor 480
- 25.3.18 Constructor(dic As Dictionary) 481
- 25.3.19 Constructor(JSONString as String) 481

25.3.21 Convert as Variant

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts JSON object to Xojo variant.

Example:

```
// build some JSON:
```

```
Var j As New JSONMBS
```

```
j.AddItemToObject "test", JSONMBS.NewNumberNode(123)
j.AddItemToObject "name", JSONMBS.NewStringNode("Joe")
```

```
// convert to Dictionary
Var v As Variant = j.Convert
```

```
// and back
Var o As JSONMBS = JSONMBS.Convert(v)
```

```
// show
MsgBox o.toString
```

Notes: Depending on the JSON content, you get variants containing values (string, double, boolean or nil), dictionaries or array of variants.

Dictionary is not case sensitive, so may not work correctly if you have duplicate keys with only different in case.

With version v24.3 we also convert JSONItem and convert folderitem, date or DateTime objects to strings. See also:

- 25.3.22 Convert(value as variant) as JSONMBS

483

25.3.22 Convert(value as variant) as JSONMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts Xojo variant to JSON object.

Example:

```
Var d As New Dictionary
Var n as Double = 1/3

d.Value("number") = n // default double handling

// custom formatted number
Var j As JSONMBS = JSONMBS.NewNumberNode(Str(n, "-0.000000000"))

d.value("customNumber") = j
d.Value( "int64") = 1000000000000000000
d.Value("uint64") = 8000000000000000000

Var a() As String
a.Append "Hello"
a.Append "World"

d.Value("array") = a
d.Value("dictionary") = New Dictionary("a":1, "b":2, "c":3)

Var r As JSONMBS = JSONMBS.Convert(d)
MsgBox r.toString
```

Notes: Converts values in variant to matching JSON structures. Currency is converted to double. Color is converted to integer. Dates are converted to string.

We detect arrays of String, Object, Variant, Single, Double, Int32, Int64, Boolean and Currency and convert them to JSON arrays. Dictionaries are converted to JSON objects with texts for keys. Anything else can raise an exception about an unsupported type.

Please note that Xojo variants don't make a difference between signed and unsigned integers, so we always convert using signed integers.

With version v19.3, the array/dictionary can also contain JSONMBS objects.

With version v24.3 we also convert JSONItem and convert folderitem, date or DateTime objects to strings. See also:

- 25.3.21 Convert as Variant

482

25.3.23 DeleteItem(index as Integer)

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Deletes an item from an array/object by index.

Example:

```
Var j As New JSONMBS
```

```
j.AddItemToObject "Hello", j.NewStringNode("Testing")
j.AddItemToObject "World", j.NewStringNode("Another node")
```

```
MsgBox j.toString
```

```
// delete named item
j.DeleteItem "Hello"
```

```
MsgBox j.toString
```

```
// delete first item
j.DeleteItem 0
```

```
MsgBox j.toString
```

Notes: Index is zero based.

See also:

- 25.3.24 DeleteItem(label as string)

484

25.3.24 DeleteItem(label as string)

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Deletes a named item from an object.

Example:

```
Var j As New JSONMBS
```

```
j.AddItemToObject "Hello", j.NewStringNode("Testing")
j.AddItemToObject "World", j.NewStringNode("Another node")
```

```
MsgBox j.toString
```

```
// delete named item
j.DeleteItem "Hello"
```

```
MsgBox j.toString
```

```
// delete first item
j.DeleteItem 0
```

```
MsgBox j.toString
```

See also:

- 25.3.23 DeleteItem(index as Integer)

25.3.25 Entries as JSONEntryMBS()

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries all entries.

Example:

```
Var j As New JSONMBS
```

```
j.Value("Hello") = "World"
j.Value("Test") = "Value"
j.Value("abc") = "cde"
```

```
Var entries() As JSONEntryMBS = j.Entries
```

Break

Notes: Convenience function to get all entries.
Converts values to variants as needed.

Works for both objects and arrays.

25.3.26 Equals(Other as JSONMBS) as Boolean

Plugin Version: 19.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if two JSON are equal.

Example:

```
Var j1 as JSONMBS = JSONMBS.NewNumberNode(5)
Var j2 as JSONMBS = JSONMBS.NewNumberNode(5)
Var j3 as JSONMBS = JSONMBS.NewNumberNode(6)
```

```
if not j1.Equals(j2) then
break // failed
end if
```

```
if j3.Equals(j2) then
break // failed
end if
```

```
Break // okay
```

Notes: We compare recursively all nodes.

If structure and values are the same, we return true, otherwise false.

Objects do not need to have same order for entries.

25.3.27 FilterObjectArray(Name as String, Other as JSONMBS) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Filters JSON object array to find matching entries.

Example:

```
Var j As JSONMBS = JSONMBS.NewArrayNode
```

```
Var j1 As JSONMBS = JSONMBS.NewObjectNode
j1.AddItemToObject "id", JSONMBS.NewNumberNode(122)
j1.AddItemToObject "name", JSONMBS.NewStringNode("John")
j.AddItemToArray j1
```

```
Var j2 As JSONMBS = JSONMBS.NewObjectNode
j2.AddItemToObject "id", JSONMBS.NewNumberNode(123)
j2.AddItemToObject "name", JSONMBS.NewStringNode("Matt")
j.AddItemToArray j2
```

```
Var j3 As JSONMBS = JSONMBS.NewObjectNode
j3.AddItemToObject "id", JSONMBS.NewNumberNode(124)
j3.AddItemToObject "name", JSONMBS.NewStringNode("Bob")
```

```
j.AddItemToArray j3
```

```
Var resultArray As JSONMBS = j.FilterObjectArray("id", JSONMBS.NewNumberNode(123))
Var firstEntry As JSONMBS = resultArray.ArrayItem(0)
```

```
Break
```

Notes: Similar as if you loop and look for index of matching items with `FindValueInObjectArray` and transfer it yourself to a new array.

25.3.28 FindValueInArray(Other as JSONMBS, StartIndex as Integer = 0) as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries index of value in array.

Example:

```
Var j1 as JSONMBS = JSONMBS.NewIntegerArray(array(3,4,5,6,7))
Var j2 as JSONMBS = JSONMBS.NewNumberNode(5)
Var j3 as JSONMBS = JSONMBS.NewNumberNode(8)
Var j4 as JSONMBS = JSONMBS.NewStringNode("5")

Var index1 as integer = j1.FindValueInArray(j2) // we can find this
Var index2 as integer = j1.FindValueInArray(j3) // value not found
Var index3 as integer = j1.FindValueInArray(j4) // can't find string in number array

if index1 = 2 and index2 = -1 and index3 = -1 then
Break // okay
else
Break // failed
end if
```

Notes: Returns zero based index or -1 if not found.

Version 20.0 or newer allows with `ByContent` parameter = true to find by content, so number can be found via text.

`StartIndex` parameter added in version 21.5: Index of first element to check. Zero if not specified. If you like to continue searching, you can pass last result + 1.

25.3.29 FindValueInObjectArray(Name as String, Other as JSONMBS, StartIndex as Integer = 0) as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries index of value in object array.

Example:

```

Var j1 as JSONMBS = JSONMBS.NewArrayNode
Var j2 as JSONMBS = JSONMBS.NewObjectNode

j2.AddItemToObject "Hello", JSONMBS.NewStringNode("World")
j2.AddItemToObject "ID", JSONMBS.NewStringNode("123")

j1.AddItemToArray j2

Var index1 as integer = j1.FindValueInObjectArray("ID", JSONMBS.NewStringNode("123")) // we can
find this
Var index2 as integer = j1.FindValueInObjectArray("ID", JSONMBS.NewStringNode("456")) // value not
found
Var index3 as integer = j1.FindValueInObjectArray("xxx", JSONMBS.NewStringNode("123")) // value not
found

if index1 = 0 and index2 = -1 and index3 = -1 then
Break // okay
else
Break // failed
end if

```

Notes: Returns zero based index or -1 if not found.

We look into each object in the array, check if it has a value for the given label and compare that to the one to find.

Version 20.0 or newer allows with ByContent parameter = true to find by content, so number can be found via text.

StartIndex parameter added in version 21.5: Index of first element to check. Zero if not specified.

If you like to continue searching, you can pass last result + 1.

25.3.30 Flatten(value as JSONMBS) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Flattens a json object or array to a single depth object of key-value pairs.

Example:


```

Var j As New JSONMBS

j.Value("Hello") = "World"
j.Value("test") = 123

Var f As JSONMBS = JSONMBS.Flatten(j)
Var s As String = f.toString

MessageBox s
// shows: { "$ [ 'Hello' ] ":"World", "$ [ 'test' ] ":123 }

```

Notes: The keys in the flattened object are normalized json paths. The values are primitive (string, number, boolean, or null), empty object ({ }) or empty array ([]).

25.3.31 HasChild(label as string) as Boolean

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if a child node for the node with the given label exists.

Example:

```

Var jv as JSONMBS = JSONMBS.NewStringNode("value")
Var jo as JSONMBS = JSONMBS.NewObjectNode

jo.AddItemToObject("key", jv)

// shows { "key": "value" }
MsgBox jo.toString
MsgBox str(jo.hasChild("key"))

```

Notes: Returns true if the entry exists or false if not.
Only for objects, not arrays.

Same as HasKey and HasName methods.

25.3.32 HasKey(Key as string) as boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if a an object has the given key.

Example:

```

Var jv as JSONMBS = JSONMBS.NewStringNode("value")
Var jo as JSONMBS = JSONMBS.NewObjectNode

```

```
jo.AddItemToObject("key", jv)

// shows { "key": "value" }
MsgBox jo.toString
MsgBox str(jo.HasKey("key"))
```

Notes: Returns true if the entry exists or false if not.
Only for objects, not arrays.

Same as HasChild and HasName methods.

25.3.33 HasName(Name as string) as boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if a an object has the given key.

Example:

```
Var jv as JSONMBS = JSONMBS.NewStringNode("value")
Var jo as JSONMBS = JSONMBS.NewObjectNode
```

```
jo.AddItemToObject("key", jv)

// shows { "key": "value" }
MsgBox jo.toString
MsgBox str(jo.HasName("key"))
```

Notes: Returns true if the entry exists or false if not.
Only for objects, not arrays.

Same as HasKey and HasChild methods.

25.3.34 Insert(index as integer, value as variant)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a value into the index.

Example:

```
Var o As New JSONMBS
```

```
o.Append 1
```

o.Insert 1,2

MessageBox o.toString

o.Insert 1,3

MessageBox o.toString

Notes: Variant is converted to JSONMBS if needed.

If the self is an empty object, we replace it with an empty array and insert the value.

Same as AddAt method.

25.3.35 isValidJSON(JSON as string) as Boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if JSON text is valid.

Notes: This parses the JSON text and returns true on success or false on failure.

This checks if the given text is a valid JSON fragment and can be e.g. added as value to a JSON object or array. So this returns true for proper formatted boolean, number and text pieces, too.

25.3.36 Iterate as JSONIteratorMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Iterate about child nodes.

Example:

```
Var o As New JSONMBS
```

```
o.add 1
```

```
o.add 2
```

```
o.add 3
```

```
For Each v As JSONMBS In o.Iterate
```

```
Break
```

```
// watch in debugger
```

```
Next
```

Notes: Provides JSONMBS objects for inspection.

Warning: If you iterate while the JSONMBS is changed you may skip some entries or get duplicates, so please avoid editing it while iterating.

25.3.37 IterateEntries as JSONIteratorMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Iterate about child entries.

Example:

```
Var o As New JSONMBS
```

```
o.add 1  
o.add 2  
o.add 3
```

```
For Each v As JSONEntryMBS In o.IterateEntries  
Break  
// watch in debugger  
Next
```

Notes: Entries are provided with JSONEntryMBS objects and include key and value. For arrays, the key name is the index value.

Warning: If you iterate while the JSONMBS is changed you may skip some entries or get duplicates, so please avoid editing it while iterating.

25.3.38 IterateValues as JSONIteratorMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Iterate about child values.

Example:

```
Var o As New JSONMBS
```

```
o.add 1  
o.add 2  
o.add 3
```

```
For Each v As Variant In o.IterateValues  
Break
```

```
// watch in debugger
// v has values like 1, 2 and 3
Next
```

Notes: The values are provided and converted to variant.

Warning: If you iterate while the JSONMBS is changed you may skip some entries or get duplicates, so please avoid editing it while iterating.

25.3.39 JSONObjectCount as Integer

Plugin Version: 15.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: For debugging, the plugin counts how many JSONMBS objects we have.

25.3.40 KeyAt(index As Integer) as String

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries key name at given index.

Example:

```
Var o As New JSONMBS
```

```
o.Value("Hello") = "World"
```

```
MessageBox o.KeyAt(0)
```

Notes: Raises an exception if index is out of range.

Same as Name or NameAt methods.

The order of keys can change if you add/remove values in the JSON object.

25.3.41 Keys as String()

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the key names for an object.

Example:

```
Var o As New JSONMBS
```

```
o.Value("Hello") = "World"  
o.Value("Test") = "Value"
```

```
Var Keys() As String = o.Keys  
MessageBox string.FromArray(Keys)
```

Notes: The order is as stored in memory currently and that order changes if you edit the JSON object. We cache the array so multiple calls would return the same array and you should not modify it.

Same as Names method.

25.3.42 Load(JSONString as String)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Parses the given JSON String into the tree where this node is the root.

Example:

```
Var o As New JSONMBS  
o.load(" { ""text"":""Hello World"" } ")  
MsgBox o.toString
```

Notes: Text should be UTF-8.
Raises an exception in case of error.

25.3.43 Lookup(Key As String, defaultValue As Variant = nil) as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Looks up the value for the key.

Notes: Variant is converted to JSONMBS if needed.

Returns default value if key is not found.

25.3.44 Merge(Other as JSONMBS)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Merges two JSON objects or arrays.

Example:

```
// merge two arrays
```

```

Var j1 As New JSONMBS(" [ 1,2,3 ] ")
Var j2 As New JSONMBS(" [ 4,2,5 ] ")

j1.Merge(j2)

MessageBox j1.toString(False)

// merge two objects
Var j3 As New JSONMBS(" { ""a"": 1, ""b"": [ 1 ] } ")
Var j4 As New JSONMBS(" { ""c"": 3, ""b"": [ 2 ] } ")

j3.Merge(j4)

MessageBox j3.toString(True)

```

Notes: For arrays, we check if the value is already there and if not, we add it. For objects, we check the values for each key. If it doesn't exist, we copy it over. If it exists, we check arrays and objects again, but other values just get replaced.

Raises exception if you try to merge object with array.

25.3.45 MergePatchFromDiff(source as JSONMBS, target as JSONMBS) as JSONMBS

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Create a JSON Merge Patch from a diff of two json documents.

Example:

```

Var source As New JSONMBS(" { "+_
" ""title"": ""Goodbye!"" ,"+_
" ""author"": { "+_
" ""givenName"": ""John"" ,"+_
" ""familyName"": ""Doe"""+_
" } ,"+_
" ""tags"": [ ""example"", ""sample"" ] ,"+_
" ""content"": ""This will be unchanged"""+_
" } ")

Var target As New JSONMBS(" { "+_
" ""title"": ""Hello!"" ,"+_
" ""author"": { "+_
" ""givenName"": ""John"""+_
" } ,"+_
" ""tags"": [ "+_

```

```

" "example""+_
" ],"+_
" "content": "This will be unchanged"+_
" "phoneNumber": "\u002B01-123-456-7890"+_
" } ")

```

```
Var patch As JSONMBS = JSONMBS.MergePatchFromDiff(source, target)
```

```
Var sourceText As String = source.toString(True)
```

```
Var targetText As String = target.toString(True)
```

```
Var PatchText As String = Patch.toString(True)
```

Break

Notes: The mergepatch function implement the IETF standard JSON Merge Patch Returns a JSON Merge Patch.

25.3.46 Name(index As Integer) as String

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries key name at given index.

Example:

```
Var o As New JSONMBS
```

```
o.Value("Hello") = "World"
```

```
MessageBox o.Name(0)
```

Notes: Raises an exception if index is out of range.

Same as NameAt or KeyAt methods.

See also:

- 25.3.109 Name as String

523

25.3.47 NameAt(index As Integer) as String

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries key name at given index.

Example:

```
Var o As New JSONMBS
```



```
o.Value("Hello") = "World"
```

```
MessageBox o.NameAt(0)
```

Notes: Raises an exception if index is out of range.

Same as Name or KeyAt methods.

25.3.48 Names as String()

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the key names for an object.

Example:

```
Var o As New JSONMBS
```

```
o.Value("Hello") = "World"
```

```
o.Value("Test") = "Value"
```

```
Var Names() As String = o.Names
```

```
MessageBox string.FromArray(Names)
```

Notes: The order is as stored in memory currently and that order changes if you edit the JSON object. We cache the array so multiple calls would return the same array and you should not modify it.

Same as Keys method.

25.3.49 NewArrayNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new node for an array.

Example:

```
Var n as JSONMBS = JSONMBS.NewArrayNode
```

```
MsgBox str(n.Type)+" = "+str(n.kTypeArray)
```

25.3.50 NewBooleanNode(value as Boolean) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Notes: This is a convenience method to quickly create an array.

25.3.51 NewBoolNode(value as boolean) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new boolean node.

Example:

```
Var j as JSONMBS = JSONMBS.NewBoolNode(true)
MsgBox j.toString
```

25.3.52 NewByteStringNode(Bytes as MemoryBlock) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new node for a byte string.

Example:

```
Var m As MemoryBlock = "Hello World"
Var j As JSONMBS = JSONMBS.NewByteStringNode(m)
```

```
MessageBox j.toString
// "SGVsbG8gV29ybGQ"
```

Notes: Creates a new JSONMBS object with type = kTypeByteString.

See also:

- 25.3.53 NewByteStringNode(Bytes as ptr, Length as UInt64) as JSONMBS 498
- 25.3.54 NewByteStringNode(Bytes as String) as JSONMBS 499

25.3.53 NewByteStringNode(Bytes as ptr, Length as UInt64) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new node for a byte string.

Example:

```

Var m As MemoryBlock = "Hello World"
Var p As ptr = m
Var size as integer = m.size
Var j As JSONMBS = JSONMBS.NewByteStringNode(p,size)

```

```

MessageBox j.toString
// "SGVsbG8gV29ybGQ"

```

Notes: Creates a new JSONMBS object with type = kTypeByteString.
See also:

- 25.3.52 NewByteStringNode(Bytes as MemoryBlock) as JSONMBS 498
- 25.3.54 NewByteStringNode(Bytes as String) as JSONMBS 499

25.3.54 NewByteStringNode(Bytes as String) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new node for a byte string.

Example:

```

Var j As JSONMBS = JSONMBS.NewByteStringNode("Hello World")

```

```

MessageBox j.toString
// "SGVsbG8gV29ybGQ"

```

Notes: Creates a new JSONMBS object with type = kTypeByteString.
See also:

- 25.3.52 NewByteStringNode(Bytes as MemoryBlock) as JSONMBS 498
- 25.3.53 NewByteStringNode(Bytes as ptr, Length as UInt64) as JSONMBS 498

25.3.55 NewCurrencyNode(value as Currency) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new currency value node.

Example:

```

Var c As Currency = 152.5678

```

```

Var j As JSONMBS = JSONMBS.NewCurrencyNode(c)

```

```

MessageBox j.toString

```

Notes: Internally we store number as string with a tag marking it as big decimal, so we output it as number. If you ask for double value, we convert it of course.

25.3.56 NewDoubleArray(values() as Double) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Example:

```
Var n() as Double = array(1.0,2,3)
Var j as JSONMBS = JSONMBS.NewDoubleArray(n)
MsgBox j.toString
```

Notes: This is a convenience method to quickly create an array.

Version 17.0 and newer return empty JSON array node in case of empty string array.
Older versions returned nil.

25.3.57 NewFalseNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new false node.

Example:

```
Var n as JSONMBS = JSONMBS.NewFalseNode
MsgBox str(n.Type)+" = "+str(n.kTypeFalse)
```

Notes: This is a node which represents a boolean false value.

25.3.58 NewInt32Array(values() as Int32) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Notes: This is a convenience method to quickly create an array.

25.3.59 NewInt64Array(values() as Int64) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Notes: This is a convenience method to quickly create an array.

25.3.60 NewInt64Node(value as Int64) as JSONMBS

Plugin Version: 17.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new node based on Int64 value.

Example:

```
Var x as int64 = 92233720368547758
Var n as JSONMBS = JSONMBS.NewInt64Node(x)
```

```
MsgBox "String: "+n.ValueString+EndOfLine+"ToString: "+n.toString+EndOfLine+"Double: "+str(n.ValueDouble,"-0")+EndOfLine+"Int64: "+str(n.ValueInteger,"-0")
```

25.3.61 NewIntegerArray(values() as Integer) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Example:

```
Var n() as Integer = array(1,2,3)
Var j as JSONMBS = JSONMBS.NewIntegerArray(n)
MsgBox j.toString
```

Notes: This is a convenience method to quickly create an array.

Version 17.0 and newer return empty JSON array node in case of empty string array. Older versions returned nil.

25.3.62 NewNullNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new null node.

Example:

```
Var n as JSONMBS = JSONMBS.NewNullNode
MsgBox str(n.Type)+" = "+str(n.kTypeNull)
```

Notes: This is a node which represents a nil value.

25.3.63 NewNumberNode(value as Double) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new number node.

Example:

```
Var n as JSONMBS = JSONMBS.NewNumberNode(123)
MsgBox str(n.Type)+" = "+str(n.kTypeNumber)
```

See also:

- 25.3.64 NewNumberNode(value as string) as JSONMBS

502

25.3.64 NewNumberNode(value as string) as JSONMBS

Plugin Version: 17.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new number node with given number as text.

Example:

```
Var n as JSONMBS = JSONMBS.NewNumberNode("92233720368547758")
```

```
MsgBox "String: "+n.ValueString+EndOfLine+"ToString: "+n.toString+EndOfLine+"Double: "+str(n.ValueDouble,"-0")+EndOfLine+"Int64: "+str(n.ValueInteger,"-0")
```

Notes: This allows you to control formatting of large integer and floating point values.

See also:

- 25.3.63 NewNumberNode(value as Double) as JSONMBS

502

25.3.65 NewObjectNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new object node.

Example:

```
Var n as JSONMBS = JSONMBS.NewObjectNode
MsgBox str(n.Type)+" = "+str(n.kTypeObject)
```

25.3.66 NewStringArray(values() as string) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Example:

```
Var n() as string = array("Hello", "World")
Var j as JSONMBS = JSONMBS.NewStringArray(n)
MsgBox j.toString
```

Notes: This is a convenience method to quickly create an array.

Version 17.0 and newer return empty JSON array node in case of empty string array.
Older versions returned nil.

25.3.67 NewStringNode(value as string) as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string node.

Example:

```
// some string with single/double quote and EndOfLine
Var s as string = "Hello'World"+EndOfLine+"this is "" a test."
```

```
// make string node
Var j as JSONMBS = JSONMBS.NewStringNode(s)
```

```
// get as JSON
Var d as string = j.toString
// show
MsgBox d
```

```
// parse again
j = new JSONMBS(d)
```

```
MsgBox j.ValueString
```

25.3.68 NewTrueNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new true node.

Example:

```
Var n as JSONMBS = JSONMBS.NewTrueNode
MsgBox str(n.Type)+" = "+str(n.kTypeTrue)
```

Notes: This is a node which represents a boolean true value.

25.3.69 NewUInt32Array(values() as UInt32) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Notes: This is a convenience method to quickly create an array.

25.3.70 NewUInt64Array(values() as UInt64) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a json array object with the given values.

Notes: This is a convenience method to quickly create an array.

25.3.71 NewUInt64Node(value as UInt64) as JSONMBS

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new node based on UInt64 value.

Example:

```
Var x as UInt64 = 92233720368547758
Var n as JSONMBS = JSONMBS.NewUInt64Node(x)
```

```
MsgBox "String: "+n.ValueString+EndOfLine+"ToString: "+n.toString+EndOfLine+"Double: "+str(n.ValueDouble,"-0")+EndOfLine+"Int64: "+str(n.ValueInteger,"-0")
```


25.3.72 Operator_Compare(Other as JSONMBS) as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares two JSONMBS objects.

Example:

```
Var o As New JSONMBS
```

```
o.add 1
```

```
o.add 2
```

```
Var j As New JSONMBS
```

```
j.add 1
```

```
j.add 2
```

```
If j = o Then
```

```
  MessageBox "equal"
```

```
Else
```

```
  MessageBox "not equal"
```

```
End If
```

Notes: Called automatically by Xojo if you use =, > and < operators.
Same as Compare method.

25.3.73 Operator_Convert as Variant()

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries all values as an array.

Example:

```
Var o As New JSONMBS
```

```
o.add 1
```

```
o.add 2
```

```
// auto convert on assignment
```

```
Var values() As Variant = o
```

```
Break // see debugger
```

Notes: Convenience function to get all values.
Converts values to variants as needed.

Works for both objects and arrays.

Same as Values method.

See also:

- 25.3.74 Operator_Convert(dic As Dictionary)

506

25.3.74 Operator_Convert(dic As Dictionary)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new object with the content of the dictionary.

Example:

```
Var d As New Dictionary
d.Value(1) = 2
d.Value("Hello") = "World"
```

```
Var j as JSONMBS = d
```

```
MessageBox j.toString(False)
```

Notes: Converts all dictionary values to JSON objects internally.

See also:

- 25.3.73 Operator_Convert as Variant()

505

25.3.75 PatchFromDiff(source as JSONMBS, target as JSONMBS) as JSONMBS

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Create a JSON Patch from a diff of two json documents.

Example:

```
Var source As New JSONMBS(" { "/"": 9, "foo": "bar" } ")
Var target As New JSONMBS(" { "baz": "qux", "foo": [ "bar", "baz" ] } ")
```

```
Var Patch As JSONMBS = JSONMBS.PatchFromDiff(source, target)
```

```
Var sourceText As String = source.toString(True)
```

```
Var targetText As String = target.toString(True)
```

```
Var PatchText As String = Patch.toString(True)
```

Break

Notes: Returns a JSON Patch.

The jsonpatch functions implement the IETF standard JavaScript Object Notation (JSON) Patch.

The JSON Patch IETF standard requires that the JSON Patch method is atomic, so that if any JSON Patch operation results in an error, the target document is unchanged. The patch function implements this requirement by generating the inverse commands and building an undo stack, which is executed if any part of the patch fails.

See also:

- 25.3.76 PatchFromDiff(source as JSONMBS, target as JSONMBS, KeyToCopy as String) as JSONMBS
507

25.3.76 PatchFromDiff(source as JSONMBS, target as JSONMBS, KeyToCopy as String) as JSONMBS

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Create a JSON Patch from a diff of two json documents.

Notes: Same as normal PatchFromDiff, but copies the given primary key field from JSON entries to the diff.

See also:

- 25.3.75 PatchFromDiff(source as JSONMBS, target as JSONMBS) as JSONMBS
506

25.3.77 Query(Path as string, Options as Integer = 0) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Performs a JSON Path query.

Example:

```
Var book1 As New JSONMBS
book1.Value("title") = "Sayings Of the Century"
book1.Value("price") = 8.95
```

```
Var book2 As New JSONMBS
book2.Value("title") = "Sword Of Honour"
book2.Value("price") = 12.99
```

```
Var bookArray As New JSONMBS
bookArray.Append book1
bookArray.Append book2
```

```
Var j As New JSONMBS
```

```
j.Value("book") = bookArray
```

```
MessageBox j.toString(True)
// shows
// {
//   "book": [
//     {
//       "title": "Sayings Of the Century",
//       "price": 8.95
//     },
//     {
//       "title": "Sword Of Honour",
//       "price": 12.99
//     }
//   ]
// }
```

```
Var r as JSONMBS = j.Query("$.book [ ?(@.price <10) ] .title")
Var s As String = r.toString
```

```
MessageBox s
// shows [ "Sayings Of the Century" ]
```

Notes: Evaluates the root value against the JSONPath expression and returns an array of values or normalized path expressions.

Returns a JSON with an array containing either values or normalized path expressions matching the JSONPath expression, or an empty array if there is no match.

For options use `kPathResultOptions*` constants. Use OR to combine multiple flags.

<code>kPathResultOptionsValue</code>	Return values.
<code>kPathResultOptionsNoDuplicates</code>	Remove duplicates.
<code>kPathResultOptionsSort</code>	Sort results.
<code>kPathResultOptionsPath</code>	Return paths.

To learn more about JSONPath, please check this website:

<https://goessner.net/articles/JsonPath/>

Learn more about the JSONPath implementation here:

<https://danielaparker.github.io/JsonCons.Net/articles/JsonPath/JsonConsJsonPath.html>

See also `Replace()` to replace found values with new values.

25.3.78 Remove(Index as Integer)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes an entry by index.

Example:

```
Var o As New JSONMBS
```

```
o.add 1
o.add 2
o.add 3
```

```
MessageBox o.toString(False)
```

```
// remove middle item
o.Remove 1
```

```
MessageBox o.toString(false)
```

Notes: Works for arrays or objects.

For objects, prefer to delete by key name since removing an entry changes the order.

Same as RemoveAt method.

See also:

- 25.3.79 Remove(Key as string)

509

25.3.79 Remove(Key as string)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes the node with the given name.

Notes: Only for JSONMBS of type object.

See also:

- 25.3.78 Remove(Index as Integer)

509

25.3.80 RemoveAt(Index as Integer)

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes an entry by index.

Example:

```
Var o As New JSONMBS
```

```
o.add 1
o.add 2
o.add 3
```

```
MessageBox o.toString(False)
```

```
// remove middle item
o.RemoveAt 1
```

```
MessageBox o.toString(false)
```

Notes: Works for arrays or objects.

For objects, prefer to delete by key name since removing an entry changes the order.

Same as Remove method.

25.3.81 Replace(Path as string, NewValue as Variant) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Searches for all values that match the JSONPath expression and replaces them with the specified value.

Example:

```
Var json As String = "{ " +_
  "  "books": " +_
  " [ " +_
  " { " +_
  "   "category": "fiction", " +_
  "   "title" : "A Wild Sheep Chase", " +_
  "   "author" : "Haruki Murakami", " +_
  "   "price" : 22.72 " +_
  " } , " +_
  " { " +_
  "   "category": "fiction", " +_
  "   "title" : "The Night Watch", " +_
  "   "author" : "Sergei Lukyanenko", " +_
  "   "price" : 23.58 " +_
  " } , " +_
  " { " +_
  "   "category": "fiction", " +_
  "   "title" : "The Comedians", " +_
  "   "author" : "Graham Greene", " +_
  "   "price" : 21.99 " +_
  " } , " +_
```

```

" { " +_
"   "category": "memoir", " +_
"   "title" : "The Night Watch", " +_
"   "author" : "Phillips, David Atlee" " +_
" } " +_
" ] " +_
" } "

```

```
Var j As New JSONMBS(json)
```

```
Var n As JSONMBS = j.Replace("$.books [ ?(@.title == 'A Wild Sheep Chase') ] .price", 123.0)
```

```
MessageBox n.toString(True)
```

Notes: Throws a `JSONExceptionMBS` if `JSONPath` evaluation fails.
Returns the modified copy of the JSON.

See also `Query()` function to just search without replace.

25.3.82 Search(Path as string) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Performs a JMESPath query.

Example:

```

Var json As String = " { "locations": [ "+EndOfLine+_
" { "name": "Seattle", "state": "WA" } , "+EndOfLine+_
" { "name": "New York", "state": "NY" } , "+EndOfLine+_
" { "name": "Bellevue", "state": "WA" } , "+EndOfLine+_
" { "name": "Olympia", "state": "WA" } "+EndOfLine+_
" ] } "

```

```
Var query As String = "locations [ ?state == 'WA' ] .name | sort(@) | { WashingtonCities: join(', ', @) } "
```

```
Var j As New JSONMBS(json)
```

```
Var r As JSONMBS = j.Search(query)
```

```
MessageBox r.toString
```

```
// { "WashingtonCities": "Bellevue, Olympia, Seattle" }
```

Notes: The `jmespath` extension implements `JMESPath`. `JMESPath` is a query language for transforming JSON documents into other JSON documents. It's supported in both the AWS and Azure CLI and has libraries available in a number of languages.

To learn more about `JMESPath`, please check this website:

<https://jmespath.org>

See also `Query()` for queries with JSONPath expressions.

25.3.83 `Sort(Reverse as boolean = false)`

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sorts values in the array by values or objects by key names.

Example:

`Var o As New JSONMBS`

`o.add 3`

`o.add 4`

`o.Add 1`

`o.Sort`

`MessageBox o.toString(False) // shows 1,3,4`

Notes: Reverse can be set to true in order to reverse the order.

25.3.84 `ToHTML(NoHeader as boolean = false, CSS as string = "") as String`

Plugin Version: 18.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts JSON to HTML.

Notes: We build for you a HTML with tables for each array and object. We include values and tag rows with even/odd CSS classes.

If `NoHeader` is true, you get just the raw table without header/footer.

Anything in `CSS` parameter is inserted before the table.

Returns HTML, which can be loaded in `htmlviewer`.

Example for CSS to do even/odd line backgrounds:

```
/* CSS style to include */
"<style>
td
{
vertical-align:top;
}
```



```
.odd
{
background-color: white;
}

.even
{
background-color: #DDD;
}
</style>”
```

25.3.85 toString(formatted as boolean) as string

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Produces a JSON markup text document from a document tree.

Example:

```
Var o as JSONMBS = JSONMBS.NewObjectNode
o.AddItemToObject "text", JSONMBS.NewStringNode("Hello World")
MsgBox o.toString // shows " { "text": "Hello World" } "
```

Notes: Returns "" on any error. Lasterror is set.

See also:

- 25.3.115 toString as String

526

25.3.86 Unflatten(value as JSONMBS) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Unflattens that object back to the original json.

Example:

```
Var t As String = " { ""$ [ 'Hello' ] "" : ""World"" , ""$ [ 'test' ] "" : 123 } "
Var f As New JSONMBS(t)
Var j As JSONMBS = JSONMBS.Unflatten(f)
Var s As String = j.toString
```

```
MessageBox s
// shows: { "Hello": "World", "test": 123 }
```

Notes: The keys in the flattened object are normalized json paths. The values are primitive (string, number, boolean, or null), empty object ({ }) or empty array ([]).

25.3.87 Values as Variant()

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries all values as an array.

Notes: Convenience function to get all values.

Converts values to variants as needed.

Works for both objects and arrays.

Same as Operator_Convert method.

25.3.88 Properties

25.3.89 ArraySize as Integer

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries size of array.

Notes: For objects returns number of keys.

Same as Count property.

(Read only property)

25.3.90 ByteStringEncoding as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Which byte string encoding to use.

Example:

```
Var mem As MemoryBlock = "Hello World"
```

```
Var dic As New Dictionary
```

```
dic.Value("test") = mem
```

```
Var j As New JSONMBS(dic)
```

```
j.ByteStringEncoding = j.kByteStringEncodingBase64
```

```
MessageBox j.toString
```

```
// { "test": "SGVsbG8gV29ybGQ=" }
```

Notes: Mainly used for encoding MemoryBlocks included in Dictionaries.
Default is Base64.
(Read and Write property)

25.3.91 CaseSensitive as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to find keys with wrong case.

Example:

```
JSONMBS.CaseSensitive = False
```

```
Var f As New JSONMBS
```

```
f.value("Hello") = "World"
```

```
// finds match
```

```
Var b1 As Boolean = f.HasKey("hello")
```

```
Var v1 As Variant = f.Value("hello")
```

```
JSONMBS.CaseSensitive = True
```

```
Var b2 As Boolean = f.HasKey("Hello")
```

```
Var v2 As Variant = f.Value("Hello")
```

```
// now try with wrong case...
```

```
Var b3 As Boolean = f.HasKey("hello")
```

```
Var v3 As Variant = f.Value("hello") // ->exception
```

```
Break
```

Notes: By default JSON is case sensitive. You can have a key "a" and another key "A" with different values in an object.

If you set CaseSensitive to true, the plugin will look case insensitive to find a match.

(Read and Write property)

25.3.92 ChildNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The first child node.

Example:

```

Var o as JSONMBS = JSONMBS.NewArrayNode

o.AddItemToArray JSONMBS.NewNumberNode(1)
o.AddItemToArray JSONMBS.NewNumberNode(2)
o.AddItemToArray JSONMBS.NewNumberNode(3)
o.AddItemToArray JSONMBS.NewNumberNode(4)

Var n as JSONMBS = o.ChildNode // first child
MsgBox n.ValueString // shows 1

```

Notes: Please move to newer iterator functions instead.
(Read only property)

25.3.93 Compact as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to prefer compact output for toString property.

Example:

```

Var j As New JSONMBS
j.Value("Hello") = "World"

Var s As String = j.toString
j.compact = false
Var f As String = j.toString

Break // see difference?

```

Notes: Default is true.
Set to false to get formatted output.
Or call ToString method with parameter.
(Read and Write property)

25.3.94 Count as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries size of array.

Example:

```

Var o As New JSONMBS

```

```
o.add 1  
o.add 2  
o.add 3
```

```
MessageBox o.Count.ToString // shows 3
```

Notes: For objects returns number of keys.
Same as `ArraySize` property.
(Read only property)

25.3.95 Handle as Integer

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal handle to the JSON object.

Notes: It is possible to have two Xojo objects with the same handle pointing to the same JSON node.
(Read only property)

25.3.96 IsArray as Boolean

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this object represents an array.

Notes: Returns true if type is `kTypeArray`.
(Read only property)

25.3.97 IsBoolean as Boolean

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this object represents a boolean value.

Notes: Returns true if type is `kTypeTrue` or `kTypeFalse`.
(Read only property)

25.3.98 IsEmpty as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the value is empty.

Example:

```

Var j As New JSONMBS

Var empty1 As Boolean = j.IsEmpty // true

j.Value("hello") = "world"

Var empty2 As Boolean = j.IsEmpty // false

Break

```

Notes: Whether a string, array or object is empty and has no content. Boolean or numerical values are not empty. (Read only property)

25.3.99 IsInt32 as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this is an 64-bit integer number.

Example:

```

Var j1 As JSONMBS = JSONMBS.NewNumberNode(1234)
Var j2 As JSONMBS = JSONMBS.NewNumberNode(1234.4)
Var j3 As JSONMBS = JSONMBS.NewNumberNode(12343456789435234523)
Var j4 As JSONMBS = JSONMBS.NewNumberNode(-6578904953445453)
Var j5 As JSONMBS = JSONMBS.NewStringNode("Hello")

Var b1 As Boolean = j1.IsInt32 // okay
Var b2 As Boolean = j2.IsInt32 // has decimals
Var b3 As Boolean = j3.IsInt32 // too big
Var b4 As Boolean = j4.IsInt32 // too small
Var b5 As Boolean = j5.IsInt32 // is string

```

Break

Notes: Returns true if

- it is type number
- it is in the range for int32.
- it is not with decimal digits after the dot.

(Read only property)

25.3.100 IsInt64 as Boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this is an 64-bit integer number.

Example:

```
Var j1 As JSONMBS = JSONMBS.NewNumberNode(1234)
Var j2 As JSONMBS = JSONMBS.NewNumberNode(1234.4)
Var j3 As JSONMBS = JSONMBS.NewNumberNode(1234345673389435234523.0)
Var j4 As JSONMBS = JSONMBS.NewNumberNode(-657890493333353445453.0)
Var j5 As JSONMBS = JSONMBS.NewStringNode("Hello")
```

```
Var b1 As Boolean = j1.IsInt64 // okay
Var b2 As Boolean = j2.IsInt64 // has decimals
Var b3 As Boolean = j3.IsInt64 // too big
Var b4 As Boolean = j4.IsInt64 // too small
Var b5 As Boolean = j5.IsInt64 // is string
```

Break

Notes: Returns true if

- it is type number
- it is in the range for int64.
- it is not with decimal digits after the dot.

(Read only property)

25.3.101 IsNull as Boolean

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this object represents a NULL value.

Notes: Returns true if type is kTypeNull.

(Read only property)

25.3.102 IsNumber as Boolean

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this object represents a numeric value.

Notes: Returns true if type is int64, uint64, half or double.

If type is string, we check semantic tag and return true for big decimal, big int, big float or float 128bit, so this is a numeric type.

(Read only property)

25.3.103 IsObject as Boolean

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this object represents an object.

Notes: Returns true if type is kTypeObject.

(Read only property)

25.3.104 IsUInt32 as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this is an 32-bit unsigned integer number.

Example:

```
Var j1 As JSONMBS = JSONMBS.NewNumberNode(1234)
Var j2 As JSONMBS = JSONMBS.NewNumberNode(1234.4)
Var j3 As JSONMBS = JSONMBS.NewNumberNode(12343456789435234523)
Var j4 As JSONMBS = JSONMBS.NewNumberNode(-6578904953445453)
Var j5 As JSONMBS = JSONMBS.NewStringNode("Hello")
```

```
Var b1 As Boolean = j1.IsUInt32 // okay
Var b2 As Boolean = j2.IsUInt32 // has decimals
Var b3 As Boolean = j3.IsUInt32 // too big
Var b4 As Boolean = j4.IsUInt32 // too small
Var b5 As Boolean = j5.IsUInt32 // is string
```

Break

Notes: Returns true if

- it is type number
- it is in the range for uint32.
- it is not with decimal digits after the dot.

(Read only property)

25.3.105 IsUInt64 as Boolean

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this is an 64-bit unsigned integer number.

Example:

```
Var j1 As JSONMBS = JSONMBS.NewNumberNode(1234)
Var j2 As JSONMBS = JSONMBS.NewNumberNode(1234.4)
Var j3 As JSONMBS = JSONMBS.NewNumberNode(1234345673389435234523.0)
Var j4 As JSONMBS = JSONMBS.NewNumberNode(-657890493333353445453.0)
Var j5 As JSONMBS = JSONMBS.NewStringNode("Hello")
```

```
Var b1 As Boolean = j1.IsUInt64 // okay
Var b2 As Boolean = j2.IsUInt64 // has decimals
```

```

Var b3 As Boolean = j3.IsUInt64 // too big
Var b4 As Boolean = j4.IsUInt64 // too small
Var b5 As Boolean = j5.IsUInt64 // is string

```

Break

Notes: Returns true if

- it is type number
- it is in the range for uint64.
- it is not with decimal digits after the dot.

(Read only property)

25.3.106 LastChildNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last child node.

Example:

```

Var o as JSONMBS = JSONMBS.NewArrayNode

o.AddItemToArray JSONMBS.NewNumberNode(1)
o.AddItemToArray JSONMBS.NewNumberNode(2)
o.AddItemToArray JSONMBS.NewNumberNode(3)
o.AddItemToArray JSONMBS.NewNumberNode(4)

```

```

Var k as JSONMBS = o.LastChildNode
MsgBox k.ValueString // shows 4

```

Notes: (Read only property)

25.3.107 LastRowIndex as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries last row index.

Example:

```

Var j As New JSONMBS

```

```
j.add 1
j.add 2
j.add 3
```

```
MessageBox j.LastRowIndex.ToString // shows 2
```

Notes: Same as Count-1.
(Read only property)

25.3.108 LineLengthLimit as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The line length limit.

Example:

```
Var jj As New JSONMBS
For i As Integer = 1 To 20
jj.Append i
next
```

```
Var j As New JSONMBS
j.Compact = False
j.Value("Hello") = jj
Var s1 As String = j.toString
j.LineLengthLimit = 60
Var s2 As String = j.toString
```

```
// second text has a line break after 14
MessageBox s1+EndOfLine+s2
```

Notes: By default is 120.
(Read and Write property)

25.3.109 Name as String

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of this node.

Notes: (Read only property)

See also:

- 25.3.46 Name(index As Integer) as String

25.3.110 NewLineCharacters as String

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The new line character to use.

Example:

```
Var j As New JSONMBS
j.Compact = false
j.Value("Hello") = "World"
Var s1 As String = j.toString
j.NewLineCharacters = EndOfLine.Windows
Var s2 As String = j.toString
```

MessageBox s1.len.toString+EndOfLine+s2.len.toString // 24 vs 26

Notes: Default is EndOfLine.Unix which is Chr(10).
Can be set to others like EndOfLine.Windows if needed.
(Read and Write property)

25.3.111 NextNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The next node on the same level.

Example:

```
Var o as JSONMBS = JSONMBS.NewArrayNode

o.AddItemToArray JSONMBS.NewNumberNode(1)
o.AddItemToArray JSONMBS.NewNumberNode(2)
o.AddItemToArray JSONMBS.NewNumberNode(3)
o.AddItemToArray JSONMBS.NewNumberNode(4)

Var k as JSONMBS = o.ChildNode
while k<>nil
MsgBox k.ValueString // shows 1, 2, 3, 4

k=k.NextNode
wend
```

Notes: Please move to newer iterator functions instead.
(Read only property)

25.3.112 PreviousNode as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The previous node on the same level.

Example:

```

Var o as JSONMBS = JSONMBS.NewArrayNode

o.AddItemToArray JSONMBS.NewNumberNode(1)
o.AddItemToArray JSONMBS.NewNumberNode(2)
o.AddItemToArray JSONMBS.NewNumberNode(3)
o.AddItemToArray JSONMBS.NewNumberNode(4)

Var k as JSONMBS = o.LastChildNode
while k<>nil
MsgBox k.valuestring // shows 4, 3, 2, 1

k=k.PreviousNode
wend

```

Notes: Please move to newer iterator functions instead.
(Read only property)

25.3.113 Root as JSONMBS

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The root object.

Notes: The root node owns the references to all children.
(Read only property)

25.3.114 Tag as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The semantic tag.

Notes: This can give a value as more detailed meaning on the usage.
e.g. we often have strings to hold numbers to preserve all digits.
(Read only property)

25.3.115 toString as String

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Produces a JSON markup text document from a document tree.

Example:

```
Var o as JSONMBS = JSONMBS.NewObjectNode
```

```
o.AddItemToObject "text", JSONMBS.NewStringNode("Hello World")
```

```
MsgBox o.toString // shows " { "text": "Hello World" } "
```

Notes: Returns "" on any error. Lasterror is set.

(Read only property)

See also:

- 25.3.85 toString(formatted as boolean) as string

513

25.3.116 Type as Integer

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The type of the node.

Example:

```
Var n as JSONMBS = JSONMBS.NewNullNode
```

```
MsgBox str(n.Type)
```

Notes: (Read only property)

25.3.117 TypeName as String

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shows type of this node as string.

Example:

```
Var o as JSONMBS = JSONMBS.NewArrayNode
```

```
MsgBox o.TypeName
```

Notes: This property was added for debugging so you can see type in debugger.

(Read only property)

25.3.118 Valid as Boolean

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether node is valid.

Notes: The handle is not zero and there are no error nodes in JSON tree.
Returns true if valid, else false.

(Read only property)

25.3.119 Value as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the value of this JSON.

Example:

```
Var j1 As JSONMBS = JSONMBS.NewStringNode("Hello World")
Var v1 As Variant = j1.Value
Var j2 As JSONMBS = JSONMBS.NewNumberNode(123)
Var v2 As Variant = j2.Value
Var j3 As JSONMBS = JSONMBS.NewNumberNode("1234.567")
Var v3 As Variant = j3.Value
Var j4 As JSONMBS = JSONMBS.NewCurrencyNode(123.4567)
Var v4 As Variant = j4.Value
```

[Break](#)

Notes: Converts value to variant.

(Read only property)

See also:

- [25.3.131 Value\(index As Integer\) as Variant](#) 531
- [25.3.132 Value\(Key As String\) as Variant](#) 532

25.3.120 ValueBoolean as Boolean

Plugin Version: 17.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The boolean value of this json node.

Notes: Returns true if type is kTypeTrue, false if kTypeFalse or true if ValueInteger <>0.
(Read only property)

25.3.121 ValueByteString as MemoryBlock

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries byte string as MemoryBlock

Example:

```
Var mem As MemoryBlock = "Hello World"

// we send a MemoryBlock into our JSON
Var dic As New Dictionary
dic.Value("test") = mem

Var j As New JSONMBS(dic)
MessageBox j.toString // auto encodes it as base64

// let's query value from child()
Var jm As JSONMBS = j.Child("test")
Var m1 As MemoryBlock = jm.ValueByteString

// and query via value()
Var m2 As MemoryBlock = j.Value("test")

Break
```

Notes: Returns nil if type is not kTypeByteString.
(Read only property)

25.3.122 ValueDouble as Double

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The value of the node.

Notes: (Read only property)

25.3.123 ValueInt64 as Int64

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Int64 value of the node.

Notes: If the node is an Int64, we return it.

Otherwise we try to convert it.

May raise exception if value can't be converted.

(Read only property)

25.3.124 ValueInteger as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Integer value of the node.

Notes: If the node is an integer, we return it.

Otherwise we try to convert it.

May raise exception if value can't be converted.

(Read only property)

25.3.125 ValueString as String

Plugin Version: 13.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The value of the node.

Notes: (Read only property)

25.3.126 ValueUInt64 as UInt64

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The UInt64 value of the node.

Notes: If the node is an UInt64, we return it.

Otherwise we try to convert it.

May raise exception if value can't be converted.

(Read only property)

25.3.127 Child(index As Integer) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set child node.

Example:

```
Var o As New JSONMBS
```

```
o.add 1
```

```
o.add 2
```

```
o.add 3
```

```
o.Child(1) = JSONMBS.NewNullNode
```

```
Var v As JSONMBS = o.Child(2) // see in debugger
```

MessageBox o.toString

Notes: If the self is an empty object, we replace it with an empty array and add the value.

Same as ChildAt method
 Raises exception if index is out of range.
 (Read and Write computed property)
 See also:

- 25.3.128 Child(Key As String) as JSONMBS

530

25.3.128 Child(Key As String) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set the child node.

Notes: If the self is an empty array, we replace it with an empty object and add the value.

Raises an exception if the value is not found.

(Read and Write computed property)

See also:

- 25.3.127 Child(index As Integer) as JSONMBS

529

25.3.129 ChildAt(index As Integer) as JSONMBS

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set child node.

Example:

`Var o As New JSONMBS`

`o.add 1`

`o.add 2`

`o.add 3`

`o.ChildAt(1) = JSONMBS.NewNullNode`

`Var v As JSONMBS = o.ChildAt(2) // see in debugger`

MessageBox o.toString

Notes: If the self is an empty object, we replace it with an empty array and add the value.

Same as ChildAt method.
Raises exception if index is out of range.
(Read and Write computed property)

25.3.130 Operator_Subscript(index As Integer) as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set value at given index.

Example:

```
Var o As New JSONMBS
```

```
o(0) = "Hello"  
o(1) = "World"  
o(2) = "test"
```

```
MessageBox o.toString  
MessageBox o(1)
```

Notes: Variant is converted to JSONMBS if needed.
Index must be in range from 0 to Count. If index is equal to count, we append the value on setting.

If the self is an empty object, we replace it with an empty array and add the value.

Same as ValueAt or Operator_Subscript.
(Read and Write computed property)

25.3.131 Value(index As Integer) as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set value at given index.

Example:

```
Var o As New JSONMBS
```

```
o.value(0) = "Hello"  
o.value(1) = "World"  
o.value(2) = "test"
```

```
MessageBox o.toString  
MessageBox o.Value(1)
```

Notes: Variant is converted to JSONMBS if needed.

Index must be in range from 0 to Count. If index is equal to count, we append the value on setting.

If the self is an empty object, we replace it with an empty array and add the value.

Same as ValueAt or Operator_Subscript.

Raises exception if index is out of range.

(Read and Write computed property)

See also:

- 25.3.119 Value as Variant 527
- 25.3.132 Value(Key As String) as Variant 532

25.3.132 Value(Key As String) as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set a value in an object by key name.

Example:

```
Var c As Currency = 1.2345
```

```
Var d as Double = c
```

```
Var j As New JSONItem
```

```
j.Value("test") = d // can't use c here!
```

```
Var m As New JSONMBS
```

```
m.Value("test") = c
```

```
MessageBox j.ToString+EndOfLine+m.toString
```

```
// shows { "test":1.2344999999999999307 } vs { "test":1.2345 } since MBS stores currency exactly.
```

Notes: Variant is converted to or from JSONMBS if needed.

Return value as variant.

Raises an exception if the value is not found.

(Read and Write computed property)

See also:

- 25.3.119 Value as Variant 527
- 25.3.131 Value(index As Integer) as Variant 531

25.3.133 ValueAt(index As Integer) as Variant

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get or set value at given index.

Example:

Var o As New JSONMBS

```
o.ValueAt(0) = "Hello"
o.ValueAt(1) = "World"
o.ValueAt(2) = "test"
```

```
MessageBox o.toString
MessageBox o.ValueAt(1)
```

Notes: Variant is converted to JSONMBS if needed.

Index must be in range from 0 to Count. If index is equal to count, we append the value on setting.

If the self is an empty object, we replace it with an empty array and set the value.

Same as Value or Operator_Subscript.
Raises exception if index is out of range.
(Read and Write computed property)

25.3.134 Constants

Byte String Encoding

Constant	Value	Description
kByteStringEncodingBase64	2	Base64 encoding
kByteStringEncodingBase64URL	3	Base64 URL encoding
kByteStringEncodingHex	1	Hex encoding.
kByteStringEncodingNone	0	Fallback to default behavior.

Query Result Options

Constant	Value	Description
kPathResultOptionsNoDuplicates	1	Remove duplicates.
kPathResultOptionsPath	4	Return paths.
kPathResultOptionsSort	2	Sort results.
kPathResultOptionsValue	0	Return value.

Tags

Constant	Value	Description
kTagBase16	&h1a	Base16 (Hex)
kTagBase64	&h1b	base64
kTagBase64url	&h1c	base64 URL
kTagBigDec	7	Big decimal
kTagBigFloat	8	Big Float
kTagBigInt	6	Big Integer
kTagClamped	&h0e	clamped
kTagCode	&h14	code
kTagDateTime	2	DateTime
kTagEpochMilli	4	epoch-milli
kTagEpochNano	5	epoch-nano
kTagEpochSecond	3	epoch-second
kTagExt	&h11	ext
kTagFloat128	9	Float 128 bit
kTagId	&h12	id
kTagMultiDimColumnMajor	&h10	multi-dim-column-major
kTagMultiDimRowMajor	&h0f	multi-dim-row-major
kTagNone	0	n/a
kTagRegex	&h13	regex
kTagUndefined	1	undefined
kTagURI	&h0d	URI

Types

Constant	Value	Description
kTypeArray	8	For an array node.
kTypeBoolean	1	For a boolean node.
kTypeByteString	7	For a byte string node. Currently only used for MemoryBlocks passed via Variant.
kTypeDouble	5	For a double node.
kTypeInt64	2	For a Int64 node.
kTypeNull	0	For a null node.
kTypeObject	9	For a object node.
kTypeSingle	4	For a single node.
kTypeString	6	For a string node. Also applies to big numbers.
kTypeUInt64	3	For an uint64 node.

Chapter 26

Math

26.1 Globals

26.1.1 IsValidCreditCardNumberMBS(Number as String) as boolean

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Verifies a credit card number.

Example:

```
MsgBox "test 49927398716 gives "+str(IsValidCreditCardNumberMBS("49927398716"))+EndOfLine+_  
"test 49927398717 gives "+str(IsValidCreditCardNumberMBS("49927398717"))
```

Notes: This function implements the Luhn algorithm. This is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers, IMEI numbers, National Provider Identifier numbers in US and Canadian Social Insurance Numbers.

Returns true on success or false on failure.

If this function returns false, you can be sure the number is not valid. But if the function returns true, you may want to check the number online with some database.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr5](#)

26.1.2 CompareNumbersMBS(v1 as Variant, v2 as Variant) as Integer

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares numbers.

Example:

```

Var u As UInt64 = 12345678901234567890
Var i As Int64 = -1

```

```

If u > i Then
Break // 12345678901234567890 > -1
Else
Break // xojo says it's smaller!
End If

```

```

If CompareNumbersMBS(u,i) > 0 Then
Break // Plugin does it right
Else
Break
End If

```

Notes: The goal: compare numbers better than Xojo. Especially UInt64/Int64 edge cases.

We treat nil as 0 value and ptr as UInt64.

If first argument is string, we just compare both as string. If first argument is numeric and second argument is string, we convert second one to number.

Blog Entries

- [News from the MBS Xojo Plugins Version 20.5](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr3](#)

26.1.3 ACosHMBS(x as Double) as Double

Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the inverse hyperbolic cosine of the real argument x.

Example:

```
MsgBox str(ACosHMBS(5))
```

Notes: acosh(1) returns +0.

acosh(x) returns a NAN for x < 1.

acosh(+infinity) returns +infinity.

26.1.4 ACosMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the principal value of the arc cosine of x in the range $[0, \pi]$.

Example:

```
MsgBox str(ACosMBS(5))
```

Notes: acos(1) returns +0.

acos(x) returns a NAN for $|x| > 1$.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.5 ArithmeticShiftMBS(value as UInt64, count as Integer) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Does an arithmetic Shift of value and cares for the sign (+ or -).

Example:

```
msgbox str(ArithmeticShiftMBS(5,3))
```

```
// displays 40 which is  $5 \cdot (2^3) = 5 \cdot 8$ 
```

Notes: Always using 32bit.

26.1.6 ASinHMBS(x as Double) as Double

Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the principal value of the arc sine of x in the range $[-\pi/2, +\pi/2]$.

Example:

```
MsgBox str(ASinHMBS(5))
```

Notes: asin(+0) returns +0.

asin(x) returns a NAN for $|x| > 1$.

26.1.7 ASinMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the principal value of the arc sine of x in the range $[-\pi/2, +\pi/2]$.

Example:

```
MsgBox str(ASinMBS(5))
```

Notes: asin(+0) returns +-0.

asin(x) returns a NAN for $|x| > 1$.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.8 ATan2MBS(x as Double, y as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the principal value of the arc tangent of y/x , using the signs of both arguments to determine the quadrant of the return value.

Example:

```
MsgBox str(ATan2MBS(3,4))
```

Notes: atan2(+0, -0) returns +-pi.

atan2(+0, +0) returns +-0.

atan2(+0, x) returns +-pi for $x < 0$.

atan2(+0, x) returns +-0 for $x > 0$.

atan2(y, +0) returns -pi/2 for $y > 0$.

atan2(+y, -infinity) returns +-pi for finite $y > 0$.

atan2(+y, +infinity) returns +-0 for finite $y > 0$.

atan2(+infinity, +x) returns +-pi/2 for finite x .

atan2(+infinity, -infinity) returns $+3\pi/4$.

atan2(+infinity, +infinity) returns $+\pi/4$.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.9 ATanHMBS(x as Double) as Double

Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the inverse hyperbolic tangent of the real argument x .

Example:

```
MsgBox str(ATanHMBS(5))
```

Notes: atanh(+0) returns +-0.

atanh(+1) returns +-infinity.

atanh(x) returns a NaN for $|x| > 1$.

26.1.10 ATanMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The atan() function computes the principal value of the arc tangent of x in the range $[-\pi/2, +\pi/2]$.

Example:

```
MsgBox str(ATanMBS(5))
```

Notes: atan(+0) returns +0.
atan(+infinity) returns $+\pi/2$.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.11 BitClearMBS(value as UInt64, mask as UInt64) as UInt64

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears the bits in the mask from the value.

Example:

```
MsgBox bin(BitClearMBS(&b1111,&b0110)) // 1001
```

Notes: Does not work for 64 bit integers.

26.1.12 BitCountMBS(value as UInt64) as Integer

Plugin Version: 6.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Counts the number of bits set.

Example:

```
MsgBox str(BitCountMBS(&b101)) // 2
MsgBox str(BitCountMBS(&b10101)) // 5
MsgBox str(BitCountMBS(&b11111111111100001111)) // 16
MsgBox str(BitCountMBS(&hFFFFFFFF)) // 32
```

Notes: Does not work for 64 bit integers.

26.1.13 BitExclMBS(value as UInt64, bitNumber as Integer) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function:

Does an bitwiseAnd using the value and bitwisenot of $2^{\text{bitNumber}}$.
Or: Switches off the bit bitNumber inside value.

Example:

```
msgBox str(BitExclMBS(80,4))
' displays 64 which is 80 without 16 =  $2^4+2^5$  without  $2^4$ 
```

Notes:

Always using 64-bit.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)

26.1.14 BitInclMBS(value as UInt64, bitNumber as Integer) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function:

Does an bitwiseOr using the value and $2^{\text{bitNumber}}$
Or: Switches on the bit bitNumber inside value.

Example:

```
msgBox str(BitInclMBS(64,4))
' displays 80 which is  $64 + 16 = 64 + 2^4$ 
```

Notes:

Always using 64-bit.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)

26.1.15 BitIsSetMBS(value as UInt64, bitNumber as Integer) as Boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Tests whether a certain bit is set inside the value

Example:

```
Var bool as boolean
```

```
bool=BitIsSetMBS(80,4)
' true, because  $2^4=16$  is included in  $80=2^4+2^6$ 
```

Notes: Always using 64-bit.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)

26.1.16 BitValMBS(bitNumber as Integer) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns $2^{\text{bitNumber}}$

Example:

```
msgBox str(BitValMBS(5))
' displays 32
```

Notes: Always using 64-bit.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)

26.1.17 BitwiseDiffMBS(x as UInt64, y as UInt64) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns all bits of val1 which are not in val2.

Example:

```
msgBox str(BitwiseDiffMBS(65,80))
' displays 1, because  $65=2^0+2^5$  includes  $1=2^0$ , which is not part of  $80=2^4+2^5$ .
```

Notes: Always using 32bit.

26.1.18 BitwiseNAndMBS(x as UInt64, y as UInt64) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: negates both values and does bitwiseAnd on them.

Example:

```
msgBox str(bitwiseNandMBS(65,80))  
' displays -82
```

Notes: Always using 32bit.

26.1.19 BitwiseNOrMBS(x as UInt64, y as UInt64) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: negates both values and does bitwiseor on them.

Example:

```
msgBox str(bitwisenorMBS(65,80))  
' displays -65
```

Notes: Always using 32bit.

26.1.20 BitwiseNotMBS(value as UInt64) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Does an bitwise negation of value.

Example:

```
msgBox str(BitwiseNotMBS(5))  
' displays -6
```

Notes: Always using 32bit.

26.1.21 BitwiseRotateMBS(value as UInt64, count as Integer, offset as Integer, width as Integer) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Does an bitwise rotation of value.

Example:

```
msgBox str(BitwiseRotateMBS(5,2,0,32))  
' displays 20
```

Notes: Always using 32bit.

26.1.22 ConvertFromFloat16MBS(Number as UInt16) as Single

Plugin Version: 15.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a 16bit floating point number to a 32bit floating point number.

Example:

```
Var h1 as UInt16 = ConvertToFloat16MBS(1.0)
Var f1 as single = ConvertFromFloat16MBS(h1) // should be 1.0

Var h2 as UInt16 = ConvertToFloat16MBS(-1.0)
Var f2 as single = ConvertFromFloat16MBS(h2) // should be -1.0

Var h3 as UInt16 = ConvertToFloat16MBS(1000.0)
Var f3 as single = ConvertFromFloat16MBS(h3) // should be 1000.0

Var h4 as UInt16 = ConvertToFloat16MBS(-1000.0)
Var f4 as single = ConvertFromFloat16MBS(h4) // should be -1000.0

Var inf as single = 65504.0
Var h5 as UInt16 = ConvertToFloat16MBS(inf)
Var f5 as single = ConvertFromFloat16MBS(h5) // should be 65504

Var nan as single = sqrt(-1)
Var h6 as UInt16 = ConvertToFloat16MBS(nan)
Var f6 as single = ConvertFromFloat16MBS(h6) // should be NAN

Break // check in debugger
```

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.5pr4](#)

26.1.23 ConvertToFloat16MBS(Number as Single) as UInt16

Plugin Version: 15.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a 32bit floating point number to a 16bit floating point number.

Example:

```

Var h1 as UInt16 = ConvertToFloat16MBS(1.0)
Var f1 as single = ConvertFromFloat16MBS(h1) // should be 1.0

Var h2 as UInt16 = ConvertToFloat16MBS(-1.0)
Var f2 as single = ConvertFromFloat16MBS(h2) // should be -1.0

Var h3 as UInt16 = ConvertToFloat16MBS(1000.0)
Var f3 as single = ConvertFromFloat16MBS(h3) // should be 1000.0

Var h4 as UInt16 = ConvertToFloat16MBS(-1000.0)
Var f4 as single = ConvertFromFloat16MBS(h4) // should be -1000.0

Var inf as single = 65504.0
Var h5 as UInt16 = ConvertToFloat16MBS(inf)
Var f5 as single = ConvertFromFloat16MBS(h5) // should be 65504

Var nan as single = sqrt(-1)
Var h6 as UInt16 = ConvertToFloat16MBS(nan)
Var f6 as single = ConvertFromFloat16MBS(h6) // should be NAN

Break // check in debugger

```

Notes: This loses precision as 16bit fit less bits than 32 bit.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.5pr4](#)

26.1.24 CosHMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the hyperbolic cosine of x.

Example:

```
MsgBox str(CosHMBS(5))
```

Notes: cosh(+0) returns 1.

cosh(+infinity) returns +infinity.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.25 CosMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the cosine of x (measured in radians).

Example:

```
MsgBox str(CosMBS(5))
```

Notes: cos(+0) returns 1.

cos(+infinity) returns a NaN.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.26 CurrencyAddMBS(value1 as Currency, value2 as Currency) as Currency

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds two currency values.

Notes: This function uses 64bit integer math to avoid rounding issues.

26.1.27 CurrencyDivMBS(value1 as Currency, value2 as Integer) as Currency

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Divides currency value by integer.

Notes: Rest is ignored.

This function uses 64bit integer math to avoid rounding issues.

26.1.28 CurrencyMulMBS(value1 as Currency, value2 as Integer) as Currency

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Multiplies currency value with integer.

Example:

```

Var c1 as Currency = 1234567890.1234

// in xoyo
Var c3 as Currency = c1 * 12345

// with plugin
Var c4 as Currency = CurrencyMulMBS(c1, 12345)

// verify by 64bit math
Var c5 as Int64 = 12345678901234
Var c6 as int64 = c5 * 12345

// and plugin is correct
MsgBox str(c3)+" "+str(c4)+" "+str(c6)

```

Notes: This function uses 64bit integer math to avoid rounding issues.

26.1.29 CurrencySubMBS(value1 as Currency, value2 as Currency) as Currency

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Subtracts value2 from value1.

Notes: This function uses 64bit integer math to avoid rounding issues.

26.1.30 CurrencyValueMBS(value as string) as Currency

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Parses string into currency value.

Notes: Replacement for val() which works better with large numbers and raises exceptions on errors.

Blog Entries

- [Currency Data Type](#)
- [MBS Xojo / Real Studio Plugins, version 14.3pr9](#)
- [Currency Data Type](#)

26.1.31 DoubleToExtendedStrMBS(x as Double) as string

Platform: macOS, Targets: All.

Function: Returns the double as an 80bit Float stored inside a ten byte string.

Example:

```
Var s as string
Var d as Double
```

```
d=5
s=DoubleToExtendedStrMBS(d)
msgBox s
d=extendedStrToDoubleMBS(s)
msgBox str(d)
```

Notes: Returns "" if there is not enough memory to create the string.

26.1.32 Exp2MBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes 2^x , the base-2 exponential of x.

Example:

```
MsgBox str(Exp2MBS(5))
```

Notes: exp2(+0) return 1.

exp2(-infinity) return +0.

exp2(+infinity) return +infinity.

26.1.33 ExpMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes e^x , the base-e exponential of x.

Example:

```
MsgBox str(ExpMBS(5))
```

Notes: exp(+0) return 1.

exp(-infinity) return +0.

exp(+infinity) return +infinity.

Blog Entries

- REAL Server Plugin 0.4

26.1.34 ExtendedStrToDoubleMBS(v as string) as Double

Platform: macOS, Targets: All.

Function: Returns the 80bit Float as a double.

Example:

```
Var s as string
Var d as Double
```

```
d=5
s=DoubleToExtendedStrMBS(d)
msgBox s
d=extendedStrToDoubleMBS(s)
msgBox str(d)
```

Notes: Returns NAN (255) if the string is not valid. e.g. "". Returns always NAN on Windows.

26.1.35 FacMBS(x as Integer) as Double

Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates (value!).

Example:

```
Sub Open()
// Fill a listbox with Fac values:
Var x as Integer
Var y as Double

for x=1 to 100
listBox1.addrow format(x,"0")
y=FacMBS(x)
if y=0 then
exit
else
listBox1.cell(listBox1.lastIndex,1)=format(y,"0")
end if
next
End Sub
```

Notes: msgBox str(facMBS(5))

' displays 120 which is $1*2*3*4*5$

26.1.36 FloorMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function returns the largest integral value less than or equal to x.

Example:

```
MsgBox str(FloorMBS(5.45))
```

Notes: floor(+0) returns +0.

floor(+infinity) returns +infinity.

Blog Entries

- [REAL Server Plugin 0.4](#)

26.1.37 FRExpMBS(inputx as Double, byref expValue as Integer) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Breaks floating-point number into normalized fraction and power of 2.

Example:

```
Var x as Double = 123.456
```

```
Var y as Integer
```

```
Var r as Double = FRExpMBS(x, y)
```

```
MsgBox str(X)+" ": "+str(y)+" "+str(r)
```

Notes: This functions break the floating-point number value into a normalized fraction and an integral power of 2. They store the integer in the int object pointed to by exp.

The functions return a number x such that x has a magnitude in the interval $[1/2, 1)$ or 0, and value = $x*(2^{\text{exp}})$.

frexp(+0, exp) returns +0, and stores 0 in the object pointed to by exp.

frexp(+infinity, exp) returns +infinity, and stores an unspecified value in the object pointed to by exp.

frexp(Nan, exp) returns a Nan, and stores an unspecified value in the object pointed to by exp.

26.1.38 HiWordMBS(i as Integer) as Integer

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the high word part of the integer.

Example:

```
MsgBox hex(HiWordMBS(&H12345678)) // shows 1234
```

Notes: equal to bitwiseshiftright(i,16)

26.1.39 HypotMBS(x as Double, y as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the $\sqrt{x^2+y^2}$ without undue overflow or underflow.

Example:

```
MsgBox str(HypotMBS(3,4)) // shows 5
```

Notes: $\text{hypot}(x, y)$, $\text{hypot}(y, x)$, and $\text{hypot}(x, -y)$ are equivalent.

$\text{hypot}(x, +0)$ is equivalent to $\text{fabs}(x)$.

$\text{hypot}(+\infty, y)$ returns $+\infty$ even if y is a NaN.

26.1.40 IsFiniteMBS(x as Double) as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if the parameter is finite.

Example:

```
Var x as Double
```

```
x=1/0
if not isfiniteMBS(x) then
msgbox "the result is Infinite."
end if
```

26.1.41 IsInfMBS(x as Double) as boolean

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if the double passed in is an infinity value.

Notes: $\log_2(1)$ return +0.
 $\log_2(x)$ return a NaN for $x < 0$.
 $\log_2(+\infty)$ return $+\infty$.

26.1.45 LogicalShiftMBS(value as UInt64, count as Integer) as UInt64

Platforms: macOS, Linux, Windows, Targets: All.

Function: Does an logical Shift of value and doesn't take care for the sign (+ or -).

Example:

```
msgbox str(LogicalShiftMBS(5,3))
```

Notes: Always using 32bit.

26.1.46 LogMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the value of the logarithm of argument x to base e.

Example:

```
MsgBox str(LogMBS(5))
```

Notes: $\log(1)$ return +0.
 $\log(x)$ return a NaN for $x < 0$.
 $\log(+\infty)$ return $+\infty$.

Blog Entries

- [REAL Server Plugin 0.4](#)

26.1.47 LoWordMBS(i as Integer) as Integer

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the low word part of the integer.

Example:

```
MsgBox hex(LoWordMBS(&H12345678)) // shows 5678
```

Notes: equal to `bitwiseand(i,&HFFFF)`

26.1.48 NormInvMBS(p as Double, mu as double = 0.0, sigma as double = 1.0) as double

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Calculates NormInv function like in Excel.

Example:

```
MsgBox Str(NormInvMBS(0.56, 5, 2)) // should show 5.3019
```

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [MBS Xojo Plugins, version 21.1pr1](#)

26.1.49 PowMBS(x as Double, y as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes x raised to the power y.

Example:

```
MsgBox str(PowMBS(5,5))
```

Notes: pow(+0, y) returns +infinity for y an odd integer <0.
 pow(+0, y) returns +infinity for y <0 and not an odd integer.
 pow(+0, y) returns +0 for y an odd integer >0.
 pow(+0, y) returns +0 for y >0 and not an odd integer.
 pow(-1, +infinity) returns 1.
 pow(1, y) returns 1 for any y, even a NaN.
 pow(x, +0) returns 1 for any x, even a NaN.
 pow(x, y) returns a NaN for finite x <0 and finite non-integer y.
 pow(x, -infinity) returns +infinity for | x | <1.
 pow(x, -infinity) returns +0 for | x | >1.
 pow(x, +infinity) returns +0 for | x | <1.
 pow(x, +infinity) returns +infinity for | x | >1.
 pow(-infinity, y) returns -0 for y an odd integer <0.
 pow(-infinity, y) returns +0 for y <0 and not an odd integer.
 pow(-infinity, y) returns -infinity for y an odd integer >0.
 pow(-infinity, y) returns +infinity for y >0 and not an odd integer.
 pow(+infinity, y) returns +0 for y <0.
 pow(+infinity, y) returns +infinity for y >0.

Range errors may occur.

Blog Entries

- [REAL Server Plugin 0.4](#)

26.1.50 RandomExponentialDistributionMBS(lambda as Double) as double

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Generates a random number using exponential distribution.

Notes: Produces random non-negative floating-point values, distributed according to probability density function.

Blog Entries

- [News from the MBS Xojo Plugins Version 24.0](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.0](#)
- [MBS Xojo Plugins, version 23.6pr2](#)

26.1.51 RandomNormalDistributionMBS(Mean as Double, StdDev as Double) as double

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Generates a random number using normal distribution.

Notes: Generates random numbers according to the Normal (or Gaussian) random number distribution.

Blog Entries

- [News from the MBS Xojo Plugins Version 24.0](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.0](#)
- [MBS Xojo Plugins, version 23.6pr2](#)

Xojo Developer Magazine

- [22.3, page 10: News](#)
- [22.2, page 9: News](#)

26.1.52 RandomPoissonDistributionMBS(Mean as Integer) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Generates a random number using poisson distribution.

Notes: Produces random non-negative integer values i , distributed according to discrete probability function.

Blog Entries

- [News from the MBS Xojo Plugins Version 24.0](#)

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.0](#)
- [MBS Xojo Plugins, version 23.6pr2](#)

Xojo Developer Magazine

- [22.3, page 10: News](#)
- [22.2, page 9: News](#)

26.1.53 RoundMBS(x as Double, decimals as Integer = 0) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function returns the integral value nearest to x rounding half-way cases away from zero, regardless of the current rounding direction.

Example:

```
MsgBox str(RoundMBS(5.5))
```

```
Var lines(-1) as string
```

```
for i as Integer = -5 to 5
```

```
lines.Append str(i)+"": "+Format(RoundMBS(123456789.123456789, i), "0.0000000")
```

```
next
```

```
MsgBox join(lines,EndOfLine)
```

Notes: round(+0) returns +-0.

round(+infinity) returns +-infinity.

Blog Entries

- [MBS REALbasic Plugins, version 11.0pr14](#)
- [REAL Server Plugin 0.4](#)

26.1.54 SinHMBS(x as Double) as Double

Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the hyperbolic sine of x.

Example:

```
MsgBox str(SinHMBS(5.5))
```

Notes: sinh(+0) returns +-0.

sinh(+infinity) returns +-infinity.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.55 SinMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the sine of x (measured in radians).

Example:

```
MsgBox str(SinMBS(5.5))
```

Notes: sin(+0) returns +0.

sin(+infinity) returns a NaN.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.56 SqrtMBS(x as Double, y as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function compute the yth root of x.

Example:

```
Var r,x,y as Double
```

```
r=SqrtMBS(x,y)
// r^y=x
```

Notes: sqrt(-0) returns -0.

sqrt(x,y) returns a NaN if the root can't be calculated.

Blog Entries

- [REAL Server Plugin 0.4](#)

26.1.57 TanHMBS(x as Double) as Double

Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the hyperbolic tangent of x.

Example:

```
MsgBox str(TanHMBS(5.5))
```

Notes: tanh(+0) returns +-0.

tanh(+infinity) returns +-1.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.58 TanMBS(x as Double) as Double

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This function computes the tangent of x (measured in radians).

Example:

```
MsgBox str(TanMBS(5.5))
```

Notes: tan(+0) returns +-0.

tan(+infinity) returns a NaN.

Blog Entries

- [MBS SQLite Extension in version 1.1](#)
- [MBS SQLite Extension, version 1.1pr1](#)

26.1.59 DoubleToInt64MBS(value as Double) as Int64

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a double value to Int64.

Blog Entries

- [MBS Real Studio Plugins, version 12.1pr7](#)

26.1.60 DoubleToUInt64MBS(value as Double) as UInt64

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a double value to UInt64 correctly.

Example:

```
// set d to a very high UInt64 value  
Var d as Double = pow(256.0,8.0) - 10000
```

```
// RB will convert to Int64 here! so number is cut to 9... instead of 18...
Var u1 as UInt64 = d

// plugin does it right
Var u2 as UInt64 = DoubleToUInt64MBS(d)

MsgBox str(u1)+" "+str(u2)
```

Notes: Xojo converts UInt64 to Double with an intermediate Int64 which breaks big numbers. This function does it correctly.

Blog Entries

- [MBS Plugins 11.1 Release notes](#)
- [MBS Real Studio Plugins, version 11.1pr11](#)

26.1.61 Int64ToDoubleMBS(value as Int64) as Double

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts an Int64 to Double.

Blog Entries

- [MBS Real Studio Plugins, version 12.1pr7](#)

26.1.62 UInt64ToDoubleMBS(value as UInt64) as Double

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts an UInt64 to Double.

Example:

```
Var d as Double = 18446744073709541376
MsgBox str(d) // fails

Var e as Double = UInt64ToDoubleMBS(18446744073709541376)
MsgBox str(e) // works
```

Notes: Xojo likes to use Int64 internally when doing math with UInt64, so we added this method to fix it.

Blog Entries

- [MBS Plugins 11.1 Release notes](#)
- [MBS Real Studio Plugins, version 11.1pr11](#)

26.2 class SplineMBS

26.2.1 class SplineMBS

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This class calculates a 2D cubic spline.

Notes: The curve goes smooth through all points.

Blog Entries

- [MBS Xojo / Real Studio plug-ins in version 15.4](#)
- [MBS Xojo / Real Studio Plugins, version 15.4pr4](#)
- [MBS Xojo / Real Studio Plugins, version 15.3pr3](#)

26.2.2 Methods

26.2.3 a(index as Integer) as Double

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constant part of the coefficient.

26.2.4 b(index as Integer) as Double

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The 1st order coefficient.

26.2.5 c(index as Integer) as Double

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The 2nd order coefficient.

26.2.6 calc(x as Double) as Double

Plugin Version: 15.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the Y value on the spline for a given X value.

26.2.7 Constructor(X() as Double, Y() as Double)

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Notes: Raises an exception for invalid arrays passed, e.g. different size. Calculates the spline curve coefficients and sets count property.

26.2.8 d(index as Integer) as Double

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The 3rd order coefficient.

26.2.9 x(index as Integer) as Double

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The x value for this coefficient.

26.2.10 y(index as Integer) as Double

Plugin Version: 15.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The y value for this coefficient.

26.2.11 Properties

26.2.12 count as Integer

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of coefficients.

Notes: (Read and Write property)

26.3 module SunTimesMBS

26.3.1 module SunTimesMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: A helper module to calculate sunset/sunrise times.

Example:

```
// may be an hour off due to daylight saving time

Var d as new date

Var jd as Double = SunTimesMBS.CalcJulianDate(d.day, d.Month, d.Year)

// Koblenz, Germany
const latitude = 50.356667
const longitude = 7.593889

// Miami, FL
'const latitude = 25.787778
'const longitude = -80.224167

Var sr as Double = SunTimesMBS.CalcSunriseUTC(jd, latitude, longitude)
Var ss as Double = SunTimesMBS.CalcSunsetUTC (jd, latitude, longitude)

d.gmtoffset = 0
d.hour = 0
d.minute = 0
d.second = 0

Var base as Double = d.totalseconds
d.totalseconds = sr*60 + base

MsgBox "Sunrise: "+d.longdate+" "+d.longtime

d.totalseconds = ss*60 + base

MsgBox "Sunset: "+d.longdate+" "+d.longtime
```

Notes: The example code above works well, but seems not to take into account the daylight saving time.

Blog Entries

- [MBS Real Studio Plugins, version 12.4pr3](#)

26.3.2 Methods

26.3.3 CalcJulianDate(day as Integer, month as Integer, year as Integer) as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a julian date from given date values.

26.3.4 CalcSunriseUTC(JD as Double, latitude as Double, longitude as Double) as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates sunrise time.

Notes: Returns time on date in minutes.

Use CalcJulianDate to calculate the JD parameter.

26.3.5 CalcSunsetUTC(JD as Double, latitude as Double, longitude as Double) as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates sunset time.

Notes: Returns time on date in minutes.

Use CalcJulianDate to calculate the JD parameter.

Chapter 27

MemoryBlock

27.1 Globals

27.1.1 NewMemoryBlockWithBytesMBS(Data as Ptr, size as Integer) as memoryblock

Plugin Version: 15.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new memoryblock with a copy of the given data.

27.1.2 NewMemoryBlockFromPtrMBS(ptr as Integer) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a memoryblock for a given address in memory.

Notes: The memory for the ptr is not freed as you have to do this yourself (or via the library that allocated it).

27.1.3 Memoryblock2ptrMBS(mem as memoryblock) as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the memory adress of the memoryblock

27.1.4 ptr2MemoryblockMBS(Value as Integer) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a memoryblock with the bytes at position mem in memory.

27.2 class Memoryblock

27.2.1 class Memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Extends Xojo's Memoryblock class.

27.2.2 Methods

27.2.3 AddressMBS(offset as Int64 = 0) as UInt64

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the adress of the byte at offset inside the memoryblock.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr5](#)

Xojo Developer Magazine

- [11.3, page 8: News](#)

27.2.4 AddressPtrMBS(offset as Int64 = 0) as Ptr

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the adress of the byte at offset inside the memoryblock.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr5](#)

27.2.5 AndBitsMBS(Second as memoryblock, Dest as memoryblock=nil) as memoryblock

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Performs a And on the bits in the memoryblock.

Example:

```
Var m1 as MemoryBlock = NewMemoryBlock(20)
Var m2 as MemoryBlock = NewMemoryBlock(20)
Var m3 as MemoryBlock = NewMemoryBlock(20)
```

```
m1.Int32Value(0)=&b10000001
m2.Int32Value(0)=&b10000000
```

```
MsgBox bin(m1.Int32Value(0))+EndOfLine+bin(m2.Int32Value(0))+EndOfLine+bin(m3.Int32Value(0))
```

```
call m1.AndBitsMBS(m2,m3)
```

```
MsgBox bin(m1.Int32Value(0))+EndOfLine+bin(m2.Int32Value(0))+EndOfLine+bin(m3.Int32Value(0))
```

Notes: Dest is first and second memoryblock combine with a bitwiseand.
And first and second memoryblock are filled with the difference between them.

If dest is nil, a new memoryblock is created. You can speed up processing with reusing the same memoryblock in iterations. If you pass a memoryblock, the plugin does not check the size of the memoryblock.

Returns nil on any error. For example if source is a memoryblock without a known size.
See also:

- 27.2.6 AndBitsMBS(Second as memoryblock, Mask as Integer, Dest as memoryblock=nil) as memoryblock 565

27.2.6 AndBitsMBS(Second as memoryblock, Mask as Integer, Dest as memoryblock=nil) as memoryblock

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Performs a And on the bits in the memoryblock.

Example:

```
Var m1 as MemoryBlock = NewMemoryBlock(20)
Var m2 as MemoryBlock = NewMemoryBlock(20)
Var m3 as MemoryBlock = NewMemoryBlock(20)
```

```
m1.Int32Value(0)=&b11110001
m2.Int32Value(0)=&b11110000
```

```
const mask = &b11001111
```

```
MsgBox bin(m1.Int32Value(0))+EndOfLine+bin(m2.Int32Value(0))+EndOfLine+bin(m3.Int32Value(0))
```

call `m1.AndBitsMBS(m2,mask,m3)`

```
MsgBox bin(m1.Int32Value(0))+EndOfLine+bin(m2.Int32Value(0))+EndOfLine+bin(m3.Int32Value(0))
```

Notes: The mask is always 8 bit. Use the `&b` notation to specify it.

Dest is first and second memoryblock combine with a bitwiseand.

And first and second memoryblock are filled with the difference between them. Only bits set int he given mask are used.

If dest is nil, a new memoryblock is created. You can speed up processing with reusing the same memoryblock in iterations. If you pass a memoryblock, the plugin does not check the size of the memoryblock.

Returns nil on any error. For example if source is a memoryblock without a known size.

See also:

- 27.2.5 `AndBitsMBS(Second as memoryblock, Dest as memoryblock=nil) as memoryblock` 564

27.2.7 AppendMBS(other as memoryblock) as memoryblock

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a new memoryblock with the bytes of the two given memoryblocks.

Example:

```
Var m1 As MemoryBlock = "Hello "
```

```
Var m2 As MemoryBlock = "World!"
```

```
Var m3 As MemoryBlock = m1.AppendMBS(m2)
```

```
Var s As String = m3
```

```
MessageBox s
```

Notes: This function will not work if the memoryblock has an unknown size.

If one memoryblock is nil than you get a copy of the other memoryblock.

Blog Entries

- [MBS Xojo Plugins, version 25.0pr9](#)

27.2.8 BitwiseAndMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock

Plugin Version: 22.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Bitwise AND for a Memoryblock.

Example:

```

Var m1 As New MemoryBlock(555)
Var m2 As New MemoryBlock(555)
Var r As New Random

For i As Integer = 0 To 555-1
m1.Byte(i) = r.InRange(0,255)
m2.Byte(i) = r.InRange(0,255)
Next

Var mAND As MemoryBlock = m1.BitwiseAndMBS(m2)
Var mOR As MemoryBlock = m1.BitwiseOrMBS(m2)
Var mXOR As MemoryBlock = m1.BitwiseXOrMBS(m2)

// check value
For i As Integer = 0 To 555-1
If BitwiseAnd(m1.Byte(i), m2.Byte(i)) <>mAND.Byte(i) Then
Break
End If
If BitwiseOr(m1.Byte(i), m2.Byte(i)) <>mOR.Byte(i) Then
Break
End If
If BitwiseXor(m1.Byte(i), m2.Byte(i)) <>mXOR.Byte(i) Then
Break
End If

Next

Break

```

Notes: We walk over all values in current and second MemoryBlock and apply a bitwise AND to each byte and store that in the destination block.

If dest is nil, a new memoryblock is created. You can speed up processing with reusing the same memoryblock in iterations. If you pass a memoryblock, the plugin does not check the size of the memoryblock.

Returns nil on any error. For example if source is a memoryblock without a known size.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.0](#)
- [MBS Xojo Plugins, version 22.6pr1](#)
- [News from the MBS Xojo Plugins Version 22.5](#)

Xojo Developer Magazine

- [21.3, page 10: News](#)
- [21.2, page 9: News](#)

27.2.9 BitwiseOrMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock

Plugin Version: 22.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Bitwise OR for a Memoryblock.

Example:

```

Var m1 As New MemoryBlock(555)
Var m2 As New MemoryBlock(555)
Var r As New Random

For i As Integer = 0 To 555-1
  m1.Byte(i) = r.InRange(0,255)
  m2.Byte(i) = r.InRange(0,255)
Next

Var mAND As MemoryBlock = m1.BitwiseAndMBS(m2)
Var mOR As MemoryBlock = m1.BitwiseOrMBS(m2)
Var mXOR As MemoryBlock = m1.BitwiseXOrMBS(m2)

// check value
For i As Integer = 0 To 555-1
  If BitwiseAnd(m1.Byte(i), m2.Byte(i)) <>mAND.Byte(i) Then
    Break
  End If
  If BitwiseOr(m1.Byte(i), m2.Byte(i)) <>mOR.Byte(i) Then
    Break
  End If
  If BitwiseXor(m1.Byte(i), m2.Byte(i)) <>mXOR.Byte(i) Then
    Break
  End If

Next

Break

```

Notes: We walk over all values in current and second MemoryBlock and apply a bitwise OR to each byte and store that in the destination block.

If dest is nil, a new memoryblock is created. You can speed up processing with reusing the same memoryblock in iterations. If you pass a memoryblock, the plugin does not check the size of the memoryblock.

Returns nil on any error. For example if source is a memoryblock without a known size.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.0](#)
- [MBS Xojo Plugins, version 22.6pr1](#)
- [News from the MBS Xojo Plugins Version 22.5](#)

Xojo Developer Magazine

- [21.3, page 10: News](#)
- [21.2, page 9: News](#)

27.2.10 BitwiseXOrMBS(Second as MemoryBlock, Dest as MemoryBlock = nil) as MemoryBlock

Plugin Version: 22.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Bitwise XOR for a Memoryblock.

Example:

```

Var m1 As New MemoryBlock(555)
Var m2 As New MemoryBlock(555)
Var r As New Random

For i As Integer = 0 To 555-1
  m1.Byte(i) = r.InRange(0,255)
  m2.Byte(i) = r.InRange(0,255)
Next

Var mAND As MemoryBlock = m1.BitwiseAndMBS(m2)
Var mOR As MemoryBlock = m1.BitwiseOrMBS(m2)
Var mXOR As MemoryBlock = m1.BitwiseXOrMBS(m2)

// check value
For i As Integer = 0 To 555-1
  If BitwiseAnd(m1.Byte(i), m2.Byte(i)) <>mAND.Byte(i) Then
  Break
End If
  If BitwiseOr(m1.Byte(i), m2.Byte(i)) <>mOR.Byte(i) Then
  Break

```

```

End If
If BitwiseXor(m1.Byte(i), m2.Byte(i)) <>mXOR.Byte(i) Then
Break
End If

Next

Break

```

Notes: We walk over all values in current and second MemoryBlock and apply a bitwise XOR to each byte and store that in the destination block.

If dest is nil, a new memoryblock is created. You can speed up processing with reusing the same memoryblock in iterations. If you pass a memoryblock, the plugin does not check the size of the memoryblock.

Returns nil on any error. For example if source is a memoryblock without a known size.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.0](#)
- [MBS Xojo Plugins, version 22.6pr1](#)
- [News from the MBS Xojo Plugins Version 22.5](#)

27.2.11 BytesEqualMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer) as Boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if equal byte data.

Example:

```

Var m1 as MemoryBlock = NewMemoryBlock(100)
Var m2 as MemoryBlock = NewMemoryBlock(100)

// try with 2 different strings
m1.CString(0)="Hello"
m2.CString(0)="Hallo"

if m1.BytesEqualMBS(0, 100, m2, 0) then
MsgBox "equal"
else
MsgBox "not equal"
end if

// try with 2 equal strings

```

```
m1.CString(0)="Hello"
m2.CString(0)="Hello"
```

```
if m1.BytesEqualMBS(0, 100, m2, 0) then
  MsgBox "equal"
else
  MsgBox "not equal"
end if
```

Notes: Fixed in 10.1 to return true on equal bytes and false on non equal bytes. Older plugin versions returned the wrong value.

Does not check the bounds of the memoryblock, so it can crash with wrong parameters.
Returns false if one of the memoryblocks is nil.

27.2.12 BytesZeroMBS(srcOfs as Integer, numBytes as Integer) as Boolean

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if all bytes in the memoryblock in the given range are zero.

Example:

```
Var m as MemoryBlock = NewMemoryBlock(100)
```

```
if m.BytesZeroMBS(0,100) then
  MsgBox "all zero"
else
  MsgBox "error"
end if
```

```
m.Byte(50)=1
```

```
if m.BytesZeroMBS(0,100) then
  MsgBox "error"
else
  MsgBox "okay"
end if
```

Notes: Returns false on any error.

Bounds are not checked with the memoryblock, so be careful.

Blog Entries

- [MBS Xojo Plugins, version 22.4pr2](#)

27.2.13 ClearBitMBS(Bit as UInt64)

Plugin Version: 21.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears the given bit in the MemoryBlock.

Example:

```
Var m As New MemoryBlock(100)

m.SetBitMBS(123)
Var v1 As Boolean = m.IsBitSetMBS(123)
m.ClearBitMBS(123)
Var v2 As Boolean = m.IsBitSetMBS(123)
Break // see debugger
```

Notes: Sets value of the bit to zero.

Raises exception is MemoryBlock is nil.

If MemoryBlock has a known size and Bit is out of range, we raise an OutOfBoundsException.

Blog Entries

- [MBS Xojo Plugins Version 21.0 News](#)
- [MBS Xojo Plugins, version 20.6pr5](#)

27.2.14 ConvertRGB12BitTo8BitMBS(Width as Integer)

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a memoryblock with 12 bit RGB data into 8 bit RGB data.

Notes: Width is number of pixels.

Please make sure the memoryblock is 9 bytes bigger than input to avoid errors.

Plugin converts each 9 bytes (72 bits) with 2 RGB triple. So first 12 bits are red, next 12 bits green and last 12 bits blue.

Plugin removes last 4 bits.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.3pr7](#)

27.2.15 CopyBytesFromMacHandleMBS(srcHandle as Integer, numBytes as Integer, destOfs as Integer)

Platform: macOS, Targets: All.

Function: Copies bytes from Mac Handle into your memoryblock.

Notes: Same as CopyBytesFromMacPtr, only that the memory address is a so-called Handle (see documentation about the MacOS memory Manager), which is double-referenced.

27.2.16 CopyBytesFromMacPtrMBS(srcPtr as Ptr, numBytes as Integer, destOfs as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the given amount of bytes from the given address in the Mac's memory address space into the memoryBlock.

Notes: Make sure that the destination block is large enough to hold the copied bytes (if not, your application or even the whole system can crash).

27.2.17 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the specified amount of bytes into a second memoryBlock.

Example:

```
Var m as MemoryBlock
```

```
Var n as MemoryBlock
```

```
m=NewMemoryBlock(100)
```

```
n=NewMemoryBlock(100)
```

```
m.Long(0)=12345
```

```
m.CopyBytesMBS(0,4,n,0)
```

```
MsgBox str(n.long(0))
```

Notes: You must make sure that the destination block is large enough to hold the copied bytes (if not, your application or even the whole system can crash).

See also:

- 27.2.18 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destOfs as Integer)

27.2.18 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destOfs as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the specified amount of bytes inside the memoryBlock.

Notes: You must make sure that the copied bytes fit inside the block (if they don't, your application or even the whole system can crash).

See also:

- 27.2.17 CopyBytesMBS(srcOfs as Integer, numBytes as Integer, destBlk as memoryBlock, destOfs as Integer) 573

27.2.19 CopyBytesToMacHandleMBS(srcOfs as Integer, numBytes as Integer, destHandle as Integer)

Platform: macOS, Targets: All.

Function: Copies byte data from the memoryblock into the memory the handle is pointing to.

Notes: Same as CopyBytesToMacPtr, only that the memory address is a so-called Handle (see documentation about the MacOS memory Manager), which is double-referenced.

27.2.20 CopyBytesToMacPtrMBS(srcOfs as Integer, numBytes as Integer, destPtr as Ptr)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the given amount of bytes from the memoryBlock to the given address in the Mac's memory address space.

Notes: Be careful where you copy the data to - you can easily crash your computer if you write to the wrong address space.

27.2.21 CopyByteToUShortMBS(dest as memoryblock, SourceOffset as Integer, DestinationOffset as Integer, ByteCount as Integer, divisor as Integer)

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies bytes in a memoryblock to another memoryblock converting the 8bit values to 16bit values.

Notes: Common values for divisor are 256 and 257.

No bound checking. Crashes with invalid values. Optimized for special divisor values.

27.2.22 CopyNthBitsMBS(source as memoryblock, SourceOffsetBits as Integer, DestinationOffsetBits as Integer, BitCount as Integer, StepCount as Integer, NumberOfSteps as Integer) as boolean

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies every nth bits in the source memoryblock to the current memoryblock.

Notes: BitCount = the number of bits to copy from each step.

StepCount = the number of bits to not copy.

NumberOfSteps = the number of rounds to do.

There is no bound checking. The function will crash with invalid parameters!

27.2.23 CopyNthBytesMBS(source as memoryblock, SourceOffsetBytes as Integer, DestinationOffsetBytes as Integer, ByteCount as Integer, StepCount as Integer, NumberOfSteps as Integer) as boolean

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies every nth bytes in the source memoryblock to the current memoryblock.

Example:

Var s,d as memoryBlock

```
s=NewmemoryBlock(100)
```

```
d=NewmemoryBlock(100)
```

```
s.CString(0)="Hello World!"
```

```
call d.CopyNthBytesMBS(s,0,0,2,4,3)
```

```
MsgBox d.CString(0) // "Heo rl"
```

Notes: ByteCount = the number of bytes to copy from each step.

StepCount = the number of bytes to move for each step forward in source.

NumberOfSteps = the number of steps to do.

There is no bound checking. The function will crash with invalid parameters!

27.2.24 CopyUShortToByteMBS(dest as memoryblock, SourceOffset as Integer, DestinationOffset as Integer, ByteCount as Integer, divisor as Integer)

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies shorts in a memoryblock to another memoryblock converting the 16bit values to 8bit

values.

Notes: Common values for divisor are 256 and 257.

No bound checking. Crashes with invalid values. Optimized for special divisor values.

27.2.25 CRC_32ContMBS(offset as Integer, numBytes as Integer, prevCRC as UInt32) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 32bit Checksum about the provided block of data.

27.2.26 CRC_32MBS(offset as Integer, numBytes as Integer) as UInt32

Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 32bit Checksum about the provided block of data.

27.2.27 CRC_CCITTContMBS(offset as Integer, numBytes as Integer, prevCRC as UInt32) as UInt32

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided string.

27.2.28 CRC_CCITTMBS(offset as Integer, numBytes as Integer) as UInt32

Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16bit Checksum about the provided block of data.

Notes: See the text "About-CRC" for details about this Checksum things.

This function is also available for strings.

See the CRC_32 for more details on Checksums.

27.2.29 CRC_DillonMBS(bitWidth as Integer, offset as Integer, numBytes as Integer) as String

Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a 16 to 64bit Checksum about the provided block of data.

Notes: This function is also available for strings.

See the CRC_32 for more details on Checksums.

27.2.30 EndianS16_BtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS16_BtoLMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.31 EndianS16_BtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS16_BtoNMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.32 EndianS16_LtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS16_LtoBMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.33 EndianS16_LtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS16_LtoNMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.34 EndianS16_NtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS16_NtoBMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.35 EndianS16_NtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS16_NtoLMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.36 EndianS32_BtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS32_BtoLMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.37 EndianS32_BtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS32_BtoNMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.38 EndianS32_LtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS32_LtoBMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.39 EndianS32_LtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS32_LtoNMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.40 EndianS32_NtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS32_NtoBMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.41 EndianS32_NtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianS32_NtoLMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.42 EndianSwap16MBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Swaps several 16 bit integers inside a memoryblock.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianSwap16MBS(0,mem.size/2)
```

27.2.43 EndianSwap32MBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Swaps several 32 bit integers inside a memoryblock.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianSwap32MBS(0,mem.size/4)
```

27.2.44 EndianU16_BtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU16_BtoLMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.45 EndianU16_BtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU16_BtoNMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.46 EndianU16_LtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU16_LtoBMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.47 EndianU16_LtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU16_LtoNMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.48 EndianU16_NtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU16_NtoBMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.49 EndianU16_NtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU16_NtoLMBS(0,mem.size/2)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.50 EndianU32_BtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU32_BtoLMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.51 EndianU32_BtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU32_BtoNMBS(0,mem.size/4)
```

Notes: e.g.:

```
EndianS32_BtoNMBS(offset as Integer,count as Integer)
```

```
EndianU16_LtoBMBS(offset as Integer,count as Integer)
```

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.52 EndianU32_LtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU32_LtoBMBS(0,mem.size/4)
```

Notes: e.g.:

```
EndianS32_BtoNMBS(offset as Integer,count as Integer)
```

```
EndianU16_LtoBMBS(offset as Integer,count as Integer)
```

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.53 EndianU32_LtoNMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU32_LtoNMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.54 EndianU32_NtoBMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU32_NtoBMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.55 EndianU32_NtoLMBS(offset as Integer,count as Integer)

Platforms: macOS, Windows, Targets: All.

Function: Converts between BigEndian, LowEndian and Native byte encoding.

Example:

```
Var mem as memoryblock // your memoryblock
mem.EndianU32_NtoLMBS(0,mem.size/4)
```

Notes: e.g.:

EndianS32_BtoNMBS(offset as Integer,count as Integer)

EndianU16_LtoBMBS(offset as Integer,count as Integer)

Details:

S for signed or U for unsigned.

16 for short and 32 for integer.

B for BigEndian (Mac), L for LowEndian (x86) and N for the native form of the current platform.

Note that count is not the size of the block, but the count of the integers to change.

27.2.56 ExpandBitsMBS(dest as memoryblock, SourceByteCount as Integer, LowValue as Integer = 0, HighValue as Integer = 255) as boolean

Plugin Version: 11.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Expands bits into bytes.

Example:

```
Var m1 as new MemoryBlock(200)
Var m2 as new MemoryBlock(1600)
```

```
for i as Integer = 0 to 127
m1.Int8Value(i) = i
next
```

```
Var n as Integer = 128
Var b as Boolean = m1.ExpandBitsMBS(m2, n)
```

`break` // see result in debugger

Notes: Takes a bit from the source memoryblock and writes a byte for it to destination. Uses LowValue (default 0) if the bit is not set and HighValue (Default 255) if the bit is set.

Works only on x86 CPUs (no PPC).

Reads source memoryblock in 32 bit blocks and writes the destination in 32 byte blocks.

Blog Entries

- [MBS REALbasic Plugins, version 10.6pr9](#)

27.2.57 ExtractBitsMBS(Mask as Integer, Dest as memoryblock=nil) as memoryblock

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies bits from a memoryblock to another.

Example:

```
Var mask as Integer = &b01010101
```

```
Var m as MemoryBlock = "Hello World"
```

```
Var r as MemoryBlock = m.ExtractBitsMBS(mask)
```

```
MsgBox r.StringValue(0,r.size) // shows "@EDDE UEPDD"
```

Notes: The mask is always 8 bit. Use the &b notation to specify it.

If dest is nil, a new memoryblock is created. You can speed up processing with reusing the same memoryblock in iterations. If you pass a memoryblock, the plugin does not check the size of the memoryblock.

Returns nil on any error. For example if source is a memoryblock without a known size.

27.2.58 FillBytesMBS(offset as Integer, count as Integer, value as Integer)

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fills memory with a specific byte value.

Notes: Caution: No bounds checking.

27.2.59 FindByteMBS(srcOfs as Integer, numBytes as Integer, byteValue as Integer) as Integer

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds the position of the first byte which has the given value.

Example:

```

Var m as MemoryBlock = NewMemoryBlock(100)
Var n as Integer

n = m.FindByteMBS(0, 100, 0)

MsgBox str(n) // shows 0 as the byte at offset 0 is zero

m.Byte(0)=1

n = m.FindByteMBS(0, 100, 0)

MsgBox str(n) // shows 1 as the byte at offset 1 is zero

m.FillBytesMBS(0,100,5)

n = m.FindByteMBS(0, 100, 6)

MsgBox str(n) // shows -1 as there is no byte with value 6

```

Notes: Returns -1 if the byte is not found.

See also:

- 27.2.60 FindByteMBS(values() as UInt8, StartByteOffset as Integer = 0) as Integer

591

27.2.60 FindByteMBS(values() as UInt8, StartByteOffset as Integer = 0) as Integer

Plugin Version: 21.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds position of a given value.

Example:

```

Var m As MemoryBlock = "Hello World"

Var u() As UInt8
u.Append Asc("e")
u.Append Asc("o")

Var pos1 As Integer = m.FindByteMBS(u)
Var pos2 As Integer = m.FindByteMBS(u, pos1+1)

// shows 1 and 4
MsgBox "Found position: "+Str(pos1)+" and "+Str(pos2)

```

Notes: Values is an array of possible values to find.
For best performance avoid duplicates in that array.

Returns zero based byte offset.
Result is -1 if not found.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.4](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.4](#)
- [MBS Xojo Plugins, version 21.4pr1](#)

Xojo Developer Magazine

- [19.6, page 10: News](#)

See also:

- [27.2.59 FindByteMBS\(srcOfs as Integer, numBytes as Integer, byteValue as Integer\) as Integer](#) 590

27.2.61 FindBytesMBS(srcOfs as Integer, maxBytes as Integer, target as memoryBlock, targOfs as Integer, targLen as Integer) as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds some bytes from the target memoryblock inside the current memoryblock.

27.2.62 FindNotByteMBS(srcOfs as Integer, numBytes as Integer, byteValue as Integer) as Integer

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds the position of the first byte which has no the given value.

Example:

```
Var m as MemoryBlock = NewMemoryBlock(100)
Var n as Integer
```

```
n = m.FindNotByteMBS(0, 100, 0)
```

```
MsgBox str(n) // -1 as no value is not zero
```

```
m.Byte(30)=1
```



```
n = m.FindNotByteMBS(0, 100, 0)

MsgBox str(n) // shows 30 as the byte at offset 30 is not zero

m.Byte(0)=255

n = m.FindNotByteMBS(0, 100, 255)

MsgBox str(n) // shows 1 as the byte at offset 1 is not 255
```

Notes: Returns -1 if the no byte was found which has not the given value.

27.2.63 FindStringMBS(srcOfs as Integer, maxBytes as Integer, target as String) as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds a string inside a memoryblock.

27.2.64 GetBitMBS(Bit as UInt64) as Integer

Plugin Version: 21.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries value of the given bit in the MemoryBlock.

Example:

```
Var m As New MemoryBlock(100)

m.SetBitMBS(123)
Var v1 As Integer = m.GetBitMBS(123)
Break // see debugger
```

Notes: Returns 1 if bit is set or 0 if not.

Raises exception is MemoryBlock is nil.

If MemoryBlock has a known size and Bit is out of range, we raise an OutOfBoundsException.

Blog Entries

- [MBS Xojo Plugins Version 21.0 News](#)
- [MBS Xojo Plugins, version 20.6pr5](#)

27.2.65 GetStringMBS(offset as Integer, numBytes as Integer) as String

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string from the memoryblock.

Notes: With newer RB versions you may better use StringValue.

27.2.66 InvertBytesMBS(offset as Integer, count as Integer)

Plugin Version: 7.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inverts the given number of bytes in a memoryblock.

Example:

```
Var m as memoryblock
m=newmemoryblock(100)
' do something
m.InvertBytesMBS(0,100) // invert all bytes
```

Notes: Does work faster if count is a multiply of 4.

27.2.67 IsBitSetMBS(Bit as UInt64) as Boolean

Plugin Version: 21.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries value of the given bit in the MemoryBlock.

Example:

```
Var m As New MemoryBlock(100)

m.SetBitMBS(123)

Var v1 As Boolean = m.IsBitSetMBS(123)
Break // see debugger
```

Notes: Returns true if bit is set or false if not.

Raises exception is MemoryBlock is nil.

If MemoryBlock has a known size and Bit is out of range, we raise an OutOfBoundsException.

Blog Entries

- [MBS Xojo Plugins Version 21.0 News](#)
- [MBS Xojo Plugins, version 20.6pr5](#)

27.2.68 LeftMBS(length as Integer) as memoryblock

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a memoryblock with the first given number of bytes.

Notes: If length is greater than the size of the memoryblock then length is set to size.

Returns nil if length<=0 or on low memory.

This function will not work if the memoryblock has an unknown size.

27.2.69 MaxMBS(firstMem as Ptr, secondMem as Ptr, BitSize as Integer = 8, Signed as Boolean = false, offsetByte as Integer = 0, lengthBytes as Integer = 0) as boolean

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates max values of values in memory.

Example:

```
Var m1 as new MemoryBlock(12)
Var m2 as new MemoryBlock(12)
Var mr1 as new MemoryBlock(12)
Var mr2 as new MemoryBlock(12)
```

```
for i as Integer = 0 to 11
m1.UInt8Value(i) = i
m2.UInt8Value(11-i) = i
next
```

```
if mr1.minMBS(m1, m2) then
if mr2.maxMBS(m1, m2) then
```

```
MsgBox EncodeHex(m1)+" first"+EndOfLine+_
EncodeHex(m2)+" second"+EndOfLine+_
EncodeHex(mr1)+" min"+EndOfLine+_
EncodeHex(mr2)+" max"
```

```
end if
end if
```

Notes: firstMem and secondMem can be ptr or memoryblock with some values. Can be same as destination (the memoryblock the method is called at)

BitSize defines integer bit depth 8, 16, 32 or 64. Signed defines if to expect signed or unsigned integers.

offsetByte defines offset in destination memoryblock. lengthBytes is length of memory. If zero, we use size of destination memoryblock.

Passing bad parameters can easily lead to crash. Return true on success and false on failure and raises exceptions for invalid parameters.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.2pr4](#)

27.2.70 MidMBS(offset as Integer) as memoryblock

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a memoryblock with the bytes of the given memoryblock from the given offset on.

Notes: Offset is 0 based.

Returns nil if offset<0 or on low memory.

This function will not work if the memoryblock has an unknown size.

See also:

- 27.2.71 MidMBS(offset as Integer, length as Integer) as memoryblock 596

27.2.71 MidMBS(offset as Integer, length as Integer) as memoryblock

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a memoryblock with the given bytes from the middle of the memoryblock.

Notes: Offset is 0 based.

Returns nil if offset<0 or on low memory.

If length is too long then length is set to a lower value.

This function will not work if the memoryblock has an unknown size.

See also:

- 27.2.70 MidMBS(offset as Integer) as memoryblock 596

27.2.72 MinMBS(firstMem as Ptr, secondMem as Ptr, BitSize as Integer = 8, Signed as Boolean = false, offsetByte as Integer = 0, lengthBytes as Integer = 0) as boolean

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates min values of values in memory.

Example:

```
Var m1 as new MemoryBlock(24)
Var m2 as new MemoryBlock(24)
Var mr1 as new MemoryBlock(24)
Var mr2 as new MemoryBlock(24)
```

```
for i as Integer = 0 to 11
m1.UInt16Value(2*i) = i
```

```

m2.UInt16Value(2*(11-i)) = i
next

if mr1.minMBS(m1, m2, 16, false, 0, 24) then
if mr2.maxMBS(m1, m2, 16, false, 0, 24) then

MsgBox EncodeHex(m1)+" first"+EndOfLine+_
EncodeHex(m2)+" second"+EndOfLine+_
EncodeHex(mr1)+" min"+EndOfLine+_
EncodeHex(mr2)+" max"

end if
end if

```

Notes: firstMem and secondMem can be ptr or memoryblock with some values. Can be same as destination (the memoryblock the method is called at)

BitSize defines integer bit depth 8, 16, 32 or 64. Signed defines if to expect signed or unsigned integers.

offsetByte defines offset in destination memoryblock. lengthBytes is length of memory. If zero, we use size of destination memoryblock.

Passing bad parameters can easily lead to crash. Return true on success and false on failure and raises exceptions for invalid parameters.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.2pr4](#)

Xojo Developer Magazine

- [18.6, page 9: News](#)

27.2.73 MirrorBitsInBytesMBS(offsetByte as Integer, lengthByte as Integer)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Mirrors bits in each byte.

Example:

```

Var m as new MemoryBlock(12)

m.CString(0)="HelloHello"

m.MirrorBitsInBytesMBS(0,5)

MsgBox EncodingToHexMBS(m.StringValue(5,5))+" ->" +EncodingToHexMBS(m.StringValue(0,5))

```

Notes: offsetByte: where to start in the memoryblock

lengthByte: number of bytes to swap

Mirror means in this case, that if you a byte with bits 11001100, after the mirror, you have 00110011. So bit 0 and 7 exchange values. Same for 1 and 6, 2 and 5, 3 and 4.

27.2.74 MirrorBitsMBS(offsetBit as Integer, lengthBit as Integer)

Plugin Version: 7.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Mirrors the bits in a memoryblock.

Notes: e.g. "111100001" in the memoryblock would give "100001111"

This function is certainly not the fastest one, but faster than anything you can get written in Xojo.

And make sure the bounds are matched, because on memory access outside the valid ranges, the function will crash.

0 ,&\$ offsetBit <mem.size*8 and 0 ,&\$ lengthBit ,&\$ mem.size*8 - offsetBit

27.2.75 MirrorBytesMBS(offsetByte as Integer = 0, lengthByte as Integer = -1)

Plugin Version: 7.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Mirrors the bytes in the given range.

Notes: e.g. "abcdefgh" in the memoryblock would give "hgfedcba"

This function is certainly not the fastest one, but faster than anything you can get written in Xojo.

And make sure the bounds are matched, because on memory access outside the valid ranges, the function will crash.

0 ,&\$ offsetByte <mem.size and 0 ,&\$ lengthByte ,&\$ mem.size - offsetByte

If lengthByte is -1, we query memoryblock for size. If size is unknown or negative, the function does nothing.

27.2.76 MultiplyUInt16MBS(Factor as Double, offsetByte as Integer = 0, lengthBytes as Integer = 0, MultiplyOnlyOver as Integer = -1)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Multiplied UInt16 values in the memoryblock.

Example:

```
Var myMemOut3 as new MemoryBlock(500)
myMemOut3.MultiplyUInt16MBS(0.9)
```

Notes: Values bigger than 65535 are set to 65535.

lengthBytes is in bytes, so 2 times the number of values.

Factor must be >= 0.0.

Offset is the offset in bytes from the beginning of the memoryblock to start. Wrong offsets (e.g. negative) can lead to crashes. lengthBytes is optional, if not specified or zero uses the length of memoryblock.

Added MultiplyOnlyOver parameter in version 18.4. If negative, we multiply all values. If positive, we multiply only values >the given value.

Blog Entries

- [MBS Xojo Plugins, version 18.4pr7](#)

27.2.77 MultiplyUInt8MBS(Factor as Double, offsetByte as Integer = 0, lengthBytes as Integer = 0, MultiplyOnlyOver as Integer = -1)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Multiplies UInt8 values in the memoryblock.

Example:

```
Var myMemOut3 as new MemoryBlock(500)
myMemOut3.MultiplyUInt8MBS(0.9)
```

Notes: Values bigger than 255 are set to 255.

lengthBytes is in bytes, so 2 times the number of values.

Factor must be >= 0.0.

Offset is the offset in bytes from the beginning of the memoryblock to start. Wrong offsets (e.g. negative) can lead to crashes. lengthBytes is optional, if not specified or zero uses the length of memoryblock.

Added MultiplyOnlyOver parameter in version 18.4. If negative, we multiply all values. If positive, we multiply only values >the given value.

Blog Entries

- [MBS Xojo Plugins, version 18.4pr7](#)

27.2.78 RightMBS(length as Integer) as memoryblock

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a memoryblock with the given number of bytes from the right side of the memoryblock.

Example:

```
Var mem,m as MemoryBlock
mem=m.RightMBS(5)
```

Notes: If length is more than the memoryblock's size, than length is set to size.

Returns nil if length<=0 or on low memory.

This function will not work if the memoryblock has an unknown size.

27.2.79 SetBitMBS(Bit as UInt64)

Plugin Version: 21.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets value of the given bit in the MemoryBlock.

Example:

```
Var m As New MemoryBlock(100)
```

```
m.SetBitMBS(123)
```

```
Var v1 As Boolean = m.IsBitSetMBS(123)
```

```
Break // see debugger
```

Notes: Sets bit to 1.

Raises exception is MemoryBlock is nil.

If MemoryBlock has a known size and Bit is out of range, we raise an OutOfBoundsException.

See also:

- 27.2.80 SetBitMBS(Bit as UInt64, Value as Boolean) 600
- 27.2.81 SetBitMBS(Bit as UInt64, Value as Integer) 601

27.2.80 SetBitMBS(Bit as UInt64, Value as Boolean)

Plugin Version: 21.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets value of the given bit in the MemoryBlock.

Example:

```
Var m As New MemoryBlock(100)
```

```
m.SetBitMBS(123, true)
```

```
Var v1 As Boolean = m.IsBitSetMBS(123)
```

```
Break // see debugger
```

Notes: Pass true to set bit or false to clear bit.

Raises exception is MemoryBlock is nil.

If MemoryBlock has a known size and Bit is out of range, we raise an OutOfBoundsException.

See also:

27.2. CLASS MEMORYBLOCK	601
• 27.2.79 SetBitMBS(Bit as UInt64)	600
• 27.2.81 SetBitMBS(Bit as UInt64, Value as Integer)	601

27.2.81 SetBitMBS(Bit as UInt64, Value as Integer)

Plugin Version: 21.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets value of the given bit in the MemoryBlock.

Example:

```
Var m As New MemoryBlock(100)
```

```
m.SetBitMBS(123, 1)
```

```
Var v1 As Boolean = m.IsBitSetMBS(123)
```

```
Break // see debugger
```

Notes: Pass 1 to set bit or 0 to clear bit.

Raises exception is MemoryBlock is nil.

If MemoryBlock has a known size and Bit is out of range, we raise an OutOfBoundsException.

Blog Entries

- [MBS Xojo Plugins Version 21.0 News](#)
- [MBS Xojo Plugins, version 20.6pr5](#)

See also:

• 27.2.79 SetBitMBS(Bit as UInt64)	600
• 27.2.80 SetBitMBS(Bit as UInt64, Value as Boolean)	600

27.2.82 SetStringMBS(str as String, offset as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a string inside the memoryblock.

Notes: With newer RB versions you may better use StringValue.

27.2.83 SwapBytes16MBS(offset as Integer, numBytes as Integer)

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Swaps words inside the given bounds inside the memoryblock.

Example:

```

Var numbytes as Integer // number of bytes in MemoryBlock
Var m as MemoryBlock // your memoryblock
Var i as Integer

for i=0 to numbytes step 2
m.UShort(i)=EndianSwap16MBS(m.UShort(i))
next

```

27.2.84 SwapBytes32MBS(offset as Integer, numBytes as Integer)

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Swaps longs inside the given bounds inside the memoryblock.

Example:

// Test SwapBytes32MBS and SwapBytes16MBS:

```

const h11223344=&h11223344
const h22114411=&h22114433 // 16bit swap
const h44332211=&h44332211 // 32bit swap

```

```

Var m as memoryBlock
Var ok as Integer

```

```

m=NewmemoryBlock(20)
m.Long(00)=h11223344
m.Long(04)=h11223344
m.Long(08)=h11223344
m.Long(12)=h11223344
m.Long(16)=h11223344

```

```

m.SwapBytes16MBS(4,4)
m.SwapBytes32MBS(12,4)

```

```

if m.Long(00)=h11223344 then
ok=ok+1
else
MsgBox "00: "+hex(m.Long(00))
end if

```

```

if m.Long(04)=h22114411 then
ok=ok+1
else
MsgBox "04: "+hex(m.Long(04))
end if

```

```

end if

if m.Long(08)=h11223344 then
ok=ok+1
else
MsgBox "08: "+hex(m.Long(08))
end if

if m.Long(12)=h44332211 then
ok=ok+1
else
MsgBox "12: "+hex(m.Long(12))
end if

if m.Long(16)=h11223344 then
ok=ok+1
else
MsgBox "16: "+hex(m.Long(16))
end if

if ok=5 then
MsgBox "OK"
else
MsgBox "Fail"
end if

```

Notes: It should do something like this:

```

Var m as memoryBlock

for i=0 to numbytes step 2
m.long(i)=EndianSwap32MBS(m.long(i))
next

```

27.2.85 SwapBytesMBS(offset as Integer, numBytes as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Swaps bytes inside the given bounds inside the memoryblock.

Notes: Reverses the order of the bytes at the given offset and length in the memoryBlock. This is helpful to change representation of values from Little Endian (used in Windows) to Big Endian (used in Mac OS) and vice versa. Thanks to Franco Vaccari for the code of this routine.

27.2.86 Properties

27.2.87 OSTypeMBS(offset as Integer) as String

Platforms: macOS, Linux, Windows, Targets: All.

Function: Read/Write an OSType.

Notes: Interprets 4 bytes starting at the given offset as a OSType value.

(Read and Write computed property)

27.3 class MemoryBlockMBS

27.3.1 class MemoryBlockMBS

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class to hold a memoryblock.

Example:

```
Var m as new MemoryBlockMBS

if m.Create(2000) then

    // copy string into memory
    m.Memory.StringValue(0,5)="Hello"

    // and read again
    MsgBox m.Memory.StringValue(0,5)

end if
```

Notes: Xojo's memoryblocks have two bad things:

- they are limited to 1 GB on Mac OS X and Mac OS Classic
- they take a lot of time to create them
- they take a lot of real memory

Xojo allocates the memory and fills it with zeros.

The plugin in contrast uses zero filled pages to create the memoryblock which nearly take no time to create them and it does only need virtual memory until the memory is really used which saved swapping space.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.1pr2](#)
- [MemoryblockMBS benchmark](#)
- [MBS REALbasic plug-ins version 9.4](#)

27.3.2 Methods

27.3.3 Close

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

Notes: There is no need to call this method except you want to free all resources of this object now without

waiting for Xojo to do it for you.

27.3.4 Constructor

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Notes: This constructor does nothing, so you can call Create yourself.

See also:

- 27.3.5 Constructor(Mem as MemoryBlock) 606
- 27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0) 607
- 27.3.7 Constructor(Size as Int64) 607
- 27.3.8 Constructor(Str as String) 607
- 27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0) 608

27.3.5 Constructor(Mem as MemoryBlock)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new MemoryBlockMBS with content of memoryblock.

Example:

```
Var s as string = "Hello World"
```

```
Var m as MemoryBlock = s
```

```
Var x as new MemoryBlockMBS(m)
```

```
MsgBox str(x.Size)
```

Notes: Raises exception if size is invalid or no memory is available.

See also:

- 27.3.4 Constructor 606
- 27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0) 607
- 27.3.7 Constructor(Size as Int64) 607
- 27.3.8 Constructor(Str as String) 607
- 27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0) 608

27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new MemoryBlockMBS with given size and copies content of memoryblock.

Example:

```
Var s as string = "Hello World"
Var m as MemoryBlock = s
```

```
Var x as new MemoryBlockMBS(m, 9)
```

```
MsgBox str(x.Size)
```

Notes: Raises exception is size is invalid or no memory is available.

See also:

- 27.3.4 Constructor 606
- 27.3.5 Constructor(Mem as MemoryBlock) 606
- 27.3.7 Constructor(Size as Int64) 607
- 27.3.8 Constructor(Str as String) 607
- 27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0) 608

27.3.7 Constructor(Size as Int64)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new MemoryBlockMBS with given size.

Notes: Raises exception is size is invalid or no memory is available.

See also:

- 27.3.4 Constructor 606
- 27.3.5 Constructor(Mem as MemoryBlock) 606
- 27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0) 607
- 27.3.8 Constructor(Str as String) 607
- 27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0) 608

27.3.8 Constructor(Str as String)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new MemoryBlockMBS with content of string.

Example:

```
Var s as string = "Hello World"
Var x as new MemoryBlockMBS(s)
```

```
MsgBox str(x.Size)
```

Notes: Text encoding is ignored and bytes copied as they are.

Raises exception is size is invalid or no memory is available.

See also:

- 27.3.4 Constructor 606
- 27.3.5 Constructor(Mem as MemoryBlock) 606
- 27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0) 607
- 27.3.7 Constructor(Size as Int64) 607
- 27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0) 608

27.3.9 Constructor(Str as String, Size as Int64, Offset as Int64 = 0)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new MemoryBlockMBS with given size and copies content of string.

Example:

```
Var s as string = "Hello World"
Var x as new MemoryBlockMBS(s, 9)
```

```
MsgBox str(x.Size)
```

Notes: Text encoding is ignored and bytes copied as they are.

Raises exception is size is invalid or no memory is available.

See also:

- 27.3.4 Constructor 606
- 27.3.5 Constructor(Mem as MemoryBlock) 606
- 27.3.6 Constructor(Mem as MemoryBlock, Size as Int64, Offset as Int64 = 0) 607
- 27.3.7 Constructor(Size as Int64) 607
- 27.3.8 Constructor(Str as String) 607

27.3.10 Create(size as Int64) as boolean

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new memoryblock.

Example:

```

Var m as MemoryBlockMBS
Var n as MemoryBlock
Var t as Integer

const size=2200000000

// RB: up to 1 GigaByte
// Plugin: up to 2 GigaByte - 1 Byte

m=New MemoryBlockMBS

t=ticks
if m.Create(size) then
n=m.Memory

MsgBox str(ticks-t)

end if

t=ticks
n=NewMemoryBlock(size)
if n<>nil then
MsgBox str(ticks-t)
end if

```

Notes: Returns true on success and false on failure.
Size can be any positive value up to &h7FFFFFFF (=2³¹-1).

Size became 64bit to avoid integer overflows.

27.3.11 Resize(Size as Int64) as boolean

Plugin Version: 9.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tries to resize the memorybock to the new size.

Example:

```

Var m as new MemoryBlockMBS

```

```
if m.Create(1000) then
MsgBox str(m.Size)

if m.Resize(2000) then
MsgBox str(m.Size)
end if
end if
```

Notes: Returns true on success.
If false is returned, the memoryblockMBS is not touched.

If the new size is smaller, data is lost as the memoryblock is cut.
If the new size is bigger, the memoryblock is resized. Or if that does not work, a new memoryblock is created and data is copied. New bytes will be zero.

27.3.12 Properties

27.3.13 Address as Int64

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The address of the memory.

Notes: This value became 64bit to avoid integer overflows.
(Read only property)

27.3.14 Memory as Memoryblock

Plugin Version: 5.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The memoryblock to access this memory.

Notes: Keep a reference to the MemoryBlockMBS object as long as you use this memoryblock object.
(Read only property)

27.3.15 Size as Int64

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size of this memoryblock.

Notes: This value became 64bit to avoid integer overflows.
(Read only property)

27.4 class MemoryStorageMBS

27.4.1 class MemoryStorageMBS

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class to store stuff in memory outside of the app memory.

Notes: The memory is stored using mapped memory in the system address space outside of the 32bit app memory of your app.

Normally a 32-bit app can only use 2 to 4 GB of memory. Using blocks of memory outside this address space, you can easily keep 10 GB of data in memory.

Be aware that at some point creating more storages may fail due to out of memory. The system may decide to swap memory to disk if you run out of physical memory.

Please have your app check free space on boot disk regularly and avoid running out of disk space for swap!

Blog Entries

- [MBS Xojo Plugins, version 18.1pr3](#)
- [MBS Xojo Plugins, version 18.1pr2](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

27.4.2 Methods

27.4.3 Constructor(Size as Int64 = 0)

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Notes: If size >0, we allocate the memory for this size right away.

27.4.4 Destructor

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

27.4.5 MemoryValue(Offset as Int64, Assigns s as MemoryBlock)

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets part of memory storage as string.

Notes: Offset and Size are checked for range and an `OutOfBoundsException` can be raised.

Please use constructor before to create memory storage with given size.

See also:

- 27.4.10 MemoryValue as MemoryBlock 613
- 27.4.6 MemoryValue(Offset as Int64, Size as Int64) as MemoryBlock 612

27.4.6 MemoryValue(Offset as Int64, Size as Int64) as MemoryBlock

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries part of memory storage as memory block.

Notes: Offset and Size are checked for range and an `OutOfBoundsException` can be raised.

See also:

- 27.4.10 MemoryValue as MemoryBlock 613
- 27.4.5 MemoryValue(Offset as Int64, Assigns s as MemoryBlock) 612

27.4.7 StringValue(Offset as Int64, Assigns s as String)

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries part of memory storage as string.

Notes: Offset and Size are checked for range and an `OutOfBoundsException` can be raised.

Please use constructor before to create memory storage with given size.

See also:

- 27.4.13 StringValue as String 613
- 27.4.8 StringValue(Offset as Int64, Size as Int64) as String 612

27.4.8 StringValue(Offset as Int64, Size as Int64) as String

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries part of memory storage as string.

Notes: Offset and Size are checked for range and an `OutOfBoundsException` can be raised.

See also:

- 27.4.13 StringValue as String 613
- 27.4.7 StringValue(Offset as Int64, Assigns s as String) 612

27.4.9 Properties

27.4.10 MemoryValue as MemoryBlock

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The content of this memory storage as a memoryblock.

Notes: (Read and Write property)

See also:

- 27.4.5 MemoryValue(Offset as Int64, Assigns s as MemoryBlock) 612
- 27.4.6 MemoryValue(Offset as Int64, Size as Int64) as MemoryBlock 612

27.4.11 Size as Integer

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size of the data in bytes.

Notes: (Read and Write property)

27.4.12 SizeAllocated as Integer

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size of memory allocated.

Notes: (Read and Write property)

27.4.13 StringValue as String

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The content of this memory storage as a string.

Notes: (Read and Write property)

See also:

- 27.4.7 StringValue(Offset as Int64, Assigns s as String) 612
- 27.4.8 StringValue(Offset as Int64, Size as Int64) as String 612

Chapter 28

Navigation

28.1 class OpenFileDialogFileTypeMBS

28.1.1 class OpenFileDialogFileTypeMBS

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: A class for a file type for the OpenFileDialog class.

Notes: For Cocoa, only the type field is used and can be an UTI or file extension.

28.1.2 Methods

28.1.3 Close

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The destructor.

Notes: There is no need to call this method except you want to free all resources of this object now without waiting for Xojo to do it for you.

28.1.4 Properties

28.1.5 Extension as String

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The extension for this filetype.

Notes: Currently this value is only used on Windows.

Value e.g. `”*.TXT”` or `”*.TXT;*.BMP”`.

Use a value like `”*.TXT”` or to match multiple types, list multiple extensions separated by semicolons like this: `”*.TXT;*.BMP”`.

With version 8.2, we use extensions also for MacOS.
(Read and Write property)

28.1.6 Name as String

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The name of this file type.

Notes: Currently this value is only used on Windows.
(Read and Write property)

28.1.7 Type as String

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The Mac OS type code for this file type.

Notes: Currently this value is only used on Mac OS.

And check the documentation on the Navigation Manager if you use it as you will need a KIND Resource for this.

For Cocoa, only the type field is used and can be an UTI or file extension.
(Read and Write property)

28.2 class OpenFileDialogItemMBS

28.2.1 class OpenFileDialogItemMBS

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The class for an item selected via open dialog.

Notes: Currently only used for Windows.

Blog Entries

- [New in the MBS Xojo Plugins 20.0](#)
- [MBS Xojo Plugins, version 20.0pr7](#)

28.2.2 Methods

28.2.3 ReadData as String

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Reads the data.

Notes: We try to get a stream for the data from Windows and then read all available data.

28.2.4 Properties

28.2.5 Handle as Integer

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The internal object handle.

Notes: Points to an IShellItem object.

(Read and Write property)

28.2.6 Name as String

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The display name of the file.

Notes: (Read and Write property)

28.2.7 Path as String

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The file path of the file.

Notes: Not available for virtual files.

(Read and Write property)

28.2.8 URL as String

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The URL of the file.

Notes: Not available for virtual files.

(Read and Write property)

28.3 class OpenFileDialogMBS

28.3.1 class OpenFileDialogMBS

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: A class for a multiple selection open dialog.

Example:

```

Var o as OpenFileDialogMBS
Var i,c as Integer
Var f as FolderItem

o=new OpenFileDialogMBS
o.ShowHiddenFiles=true
o.PromptText="Select one or more files:"
o.MultipleSelection=true
o.ActionButtonLabel="Open files"
o.CancelButtonLabel="no, thanks."
o.WindowTitle="This is a window title."
o.ClientName="Client Name?"
o.ShowDialog

c=o.FileCount
if c>0 then
for i=0 to c-1
f=o.Files(i)

ListBox1.AddRow f.NativePath
next

else
// MsgBox "no file was selected."
quit
end if

```

Notes: OpenFileDialogMBS does not yet use NSOpenPanel yet. So for Cocoa applications you can use NSOpenPanelMBS class directly. For the Mac App Store, you must use NSOpenPanelMBS and avoid OpenFileDialogMBS.

Blog Entries

- [MBS Xojo Plugins, version 21.6pr3](#)
- [MBS Xojo Plugins, version 21.1pr8](#)
- [MBS Xojo Plugins, version 20.2pr2](#)
- [Customized File Open and Save Dialogs for Windows](#)

- [MBS Xojo Plugins, version 20.0pr7](#)
- [Open Dialog upgraded](#)
- [Xojo 2017 Release 2](#)
- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.0](#)
- [MBS REALbasic Plugins, version 11.0fc1](#)
- [MonkeyBread Software Releases the MBS Plugins 8.1](#)

Xojo Developer Magazine

- [3.2, page 42: Notes, Questions, and Answers, Christian tackles a variety of issues by Christian Schmitz](#)
- [12.2, page 10: News](#)

28.3.2 Methods

28.3.3 AddType(t as OpenFileDialogTypeMBS)

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Adds a file type to the list.

28.3.4 ClearTypes

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Clears the file type list.

28.3.5 CountTypes as Integer

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Returns the number of file types in the file type list.

28.3.6 Files as FolderItem()

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Queries an array with all files picked.

See also:

- 28.3.7 Files(index as Integer) as folderitem

28.3.7 Files(index as Integer) as folderitem

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The array with the selected files.

Notes: Index goes from 0 to filecount-1.

See also:

- 28.3.6 Files as FolderItem()

28.3.8 GetCustomImageHeight as Integer

Plugin Version: 11.0, Platform: Windows, Targets: Desktop only.

Function: Returns the height of the custom picture in pixels.

Example:

```
Var o as new OpenFileDialogMBS
```

```
msgbox "GetCustomImageHeight: "+str(o.GetCustomImageHeight) // could show 72 as value.
```

Notes: The height depends on the size of the Windows system font, so it's variable and this function calculates the size for you.

Returns 0 on any error.

28.3.9 GetType(index as Integer) as OpenFileDialogFileTypeMBS

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Returns the file type with the given index.

Notes: Returns nil on any error.

28.3.10 Items as OpenFileDialogItemMBS()

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Queries items picked.

Notes: Currently only provided for Windows to get name, path, URL or data for items, which may not always have a file path.

28.3.11 RefreshCustomImage

Plugin Version: 11.0, Platform: Windows, Targets: Desktop only.

Function: Call this method after you set the custompicture property to update the window.

Notes: The control is asked to redraw and will use the CustomPicture property for the picture content. Space around is colored with window background color.

28.3.12 ShowDialog

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Shows the dialog.

Example:

```
Var o as new OpenFileDialogMBS
o.ShowDialog
```

Notes: Check the FileCount property to see whether something was selected.

28.3.13 Properties

28.3.14 accessoryView as Variant

Plugin Version: 14.0, Platform: macOS, Targets: Desktop only.

Function: Customizes the panel for the application by adding a custom view to the panel.

Notes: Only for Cocoa target. Must be a NSViewMBS or subclass.

The custom object that is added appears just above the OK and Cancel buttons at the bottom of the panel. The open panel automatically resizes itself to accommodate the view. You can invoke this method repeatedly to change the accessory view as needed. If view is nil, the panel removes the current accessory view. (Read and Write property)

28.3.15 ActionButtonLabel as String

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Function: The label of the action button.

Example:

```
Var o as new OpenFileDialogMBS
```

```
o.ActionButtonLabel = "Hello"
o.ShowDialog
```

Notes: Used only on Mac OS.

For RB 4.5 and earlier you need to make sure the encoding is set correctly.

Can be set to "" to use the default value.

(Read and Write property)

28.3.16 AllowFolderSelection as Boolean

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: Whether folders can be selected.

Example:

```
Var o as OpenFileDialogMBS
Var i,c as Integer
Var f as FolderItem
```

```
o=new OpenFileDialogMBS
o.ShowHiddenFiles=true
o.PromptText="Select one or more files/folders:"
o.MultipleSelection=true
o.ActionButtonLabel="Open files/folders"
o.CancelButtonLabel="no, thanks."
o.WindowTitle="This is a window title."
o.ClientName="Client Name?"
o.AllowFolderSelection=true
o.ShowDialog
```

```
c=o.FileCount
if c>0 then
for i=0 to c-1
f=o.Files(i)
```

```
Listbox1.AddRow f.NativePath
next
end if
```

Notes: Default is false.

Setting this to true on Windows or Linux has no effect there.

(Read and Write property)

28.3.17 CancelButtonLabel as String

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. **Function:** The text for the cancel button.

Example:

```
Var o as new OpenFileDialogMBS
```

```
o.CancelButtonLabel = "Hello"  
o.ShowDialog
```

Notes: Used only on Mac OS Carbon.
Can be set to "" to use the default value.

Deprecated as no longer needed for Windows, Linux or MacOS Cocoa.
(Read and Write property)

28.3.18 ClientName as String

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. **Function:** The name of the client.

Example:

```
Var o as new OpenFileDialogMBS
```

```
o.ClientName = "Hello"  
o.ShowDialog
```

Notes: Used only on Mac OS Carbon.
Can be set to "" to use the default value.

Deprecated as no longer needed for Windows, Linux or MacOS Cocoa.
(Read and Write property)

28.3.19 Creator as String

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Function: The creator code of your application.

Notes: This property is only used Mac OS if you specify one or more file types.
(Read and Write property)

28.3.20 CustomPicture as Picture

Plugin Version: 11.0, Platform: Windows, Targets: Desktop only.

Function: The custom picture content.

Notes: You need to call RefreshCustomImage to refresh the picture visible to the user.
(Read and Write property)

28.3.21 File as FolderItem

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The first file selected.

Example:

```
Var o as new OpenFileDialogMBS
o.MultipleSelection = true
o.ShowDialog
```

```
MsgBox o.File.displaypathmbs
```

Notes: file = files(0)

Just for the cases where you only need the first file.
(Read only property)

28.3.22 FileCount as Integer

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The number of files selected.

Example:

```
Var o as new OpenFileDialogMBS
o.MultipleSelection = true
o.ShowDialog
```

```
MsgBox str(o.FileCount)
```

Notes: (Read only property)

28.3.23 InitialDirectory as FolderItem

Plugin Version: 4.2, Platforms: macOS, Windows, Targets: Desktop only.

Function: The initial directory.

Example:

```
Var o as new OpenFileDialogMBS
o.InitialDirectory = SpecialFolder.Desktop
o.ShowDialog
```

Notes: Set to nil to get the last directory used.
(Read and Write property)

28.3.24 Lasterror as Integer

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The last error code reported.

Notes: (Read and Write property)

28.3.25 Left as Integer

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Function: The horizontal position of the dialog.

Notes: -1 is for the default position.

Used only on the Mac side.

(Read and Write property)

28.3.26 MultipleSelection as Boolean

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Whether you want to allow multiple file selection.

Example:

```
Var o as new OpenFileDialogMBS
o.MultipleSelection = true
o.ShowDialog
```

Notes: (Read and Write property)

28.3.27 ParentWindow as Variant

Plugin Version: 6.3, Platforms: Linux, Windows, Targets: Desktop only.

Function: The parent window of the dialog.

Example:

```
Var o as new OpenFileDialogMBS
o.ParentWindow = window1
o.ShowDialog
```

Notes: Only implemented for Windows and Linux.
Can reference a Window or DesktopWindow object.
(Read and Write property)

28.3.28 PromptText as String

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Function: The prompt text.

Example:

```
Var o as new OpenFileDialogMBS

o.PromptText = "Hello"
o.ShowDialog
```

Notes: For RB 4.5 and earlier you need to make sure the encoding is set correctly.
Can be set to "" to use the default value.
(Read and Write property)

28.3.29 ResolveAliases as Boolean

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Whether aliases should be resolved.

Example:

```
Var o as new OpenFileDialogMBS
o.ResolveAliases = false
o.ShowDialog
```

Notes: This property is only used on Mac OS X.
Default is true.

Flag seems to be broken on Mac OS X 10.7.
(Read and Write property)

28.3.30 ShowHiddenFiles as Boolean

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Whether hidden files should be shown.

Example:

```
Var o as new OpenFileDialogMBS
o.ShowHiddenFiles = true
o.ShowDialog
```

Notes: Default is false.
(Read and Write property)

28.3.31 Top as Integer

Plugin Version: 4.2, Platform: macOS, Targets: Desktop only.

Function: The vertical position of the dialog.

Notes: -1 is for the default position.

Used only on the Mac side.
(Read and Write property)

28.3.32 TreatFilePackagesAsDirectories as Boolean

Plugin Version: 14.0, Platform: macOS, Targets: Desktop only.

Function: Sets the panel's behavior for displaying file packages (for example, MyApp.app) to the user.

Notes: If true, the panel will display file packages as directories; if false, it will not.

(Read and Write property)

28.3.33 UseCustomPicture as Integer

Plugin Version: 11.0, Platform: Windows, Targets: Desktop only.

Function: Whether to add a custom picture control.

Notes: Specify the size of the area you want. As the size is relative to the font size of the system font on

windows, the results can be interesting. For example a value of 50 gives here a 72 pixel height area. Value can be from 1 to 100. Or zero to disable.

This method was added for a client who needed to show a small preview of a project file in the open dialog on Windows.

For Mac OS X you can use the NSSavePanelMBS/NSOpenPanelMBS with an accessory view.
(Read and Write property)

28.3.34 WindowTitle as String

Plugin Version: 4.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The window title to use.

Example:

```
Var o as new OpenFileDialogMBS
```

```
o.WindowTitle = "Hello World"  
o.ShowDialog
```

Notes: Can be set to "" to use the default value.

Not shown on MacOS.

(Read and Write property)

28.3.35 Events

28.3.36 FilterItem(file as folderitem, filterMode as Integer) as boolean

Plugin Version: 4.2, Platforms: macOS, Windows, Targets: Desktop only.

Function: An event where you can filter the file list.

Notes: Return true to disable the file.

The result of the event is ignored by Windows. But on Mac OS it works.

So use the event on Mac and the filetype objects on Windows.

28.3.37 SelectionChanged(file as folderitem)

Plugin Version: 11.0, Platform: Windows, Targets: Desktop only.

Function: Called when the selection changes in the window.

Notes: This is currently only implemented for Windows and gives you only one file name.

Chapter 29

Process

29.1 class Application

29.1.1 class Application

Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Extends the Application class inside Xojo.

29.1.2 Methods

29.1.3 ApplicationCreatorCodeMBS as string

Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns the Creator Code of the application.

Example:

```
msgbox "Hi. This application has the creator code "+app.ApplicationCreatorCodeMBS
```

Notes: Only useful on Mac OS.

If you run your app inside the Xojo IDE it will return "RBv2" which is Xojo's Creator code since Xojo 2.0.

Blog Entries

- [MBS Xojo Plugins, version 19.4pr1](#)

29.1.4 ApplicationFileMBS as folderitem

Platforms: macOS, Windows, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use ExecutableFile instead.

Function: Gives a folderitem to the applications file.

Example:

```
msgbox "Hi. The file of this application is named "+app.ApplicationFileMBS.name
```

Notes: Inside Xojo points to the Xojo application.

This item may go in the future. Please use app.executablefile in new Xojo versions.

Blog Entries

- [MBS Plugins updated for Xojo 2019r2](#)
- [MBS Xojo Plugins, version 19.4pr1](#)

Xojo Developer Magazine

- [3.1, page 42: Register Document File Types, Fill and Query the Registry by Christian Schmitz](#)

29.1.5 ApplicationNameMBS as string

Platform: macOS, Targets: Desktop only.

Function: Returns the name of the application.

Example:

```
msgbox "Hi. This application is named "+app.ApplicationNameMBS
```

Notes: This may not be the exact same name than the filename.

29.1.6 ArgumentsMBS as String()

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Queries arguments of process.

Notes: Returns nil in case of any error (low memory).

First entry in result is path to current app.

Other entries are parameters, without any quotes.

29.1.7 BundleFolderMBS as folderitem

Platform: macOS, Targets: Desktop only.

Function: Returns a folderitem to the bundle's mail folder.

29.1.8 BundleLocalizedStringMBS(key as string) as string

Platform: macOS, Targets: Desktop only.

Function: Returns the localized string for the given key from the default table.

Notes: A short version for BundleLocalizedString which uses the default table.

See also:

- 29.1.9 BundleLocalizedStringMBS(key as string,fromtable as string) as string

633

29.1.9 BundleLocalizedStringMBS(key as string,fromtable as string) as string

Platform: macOS, Targets: Desktop only.

Function: Returns the localized string for the given key and table.

Example:

```
// You may make yourself a function to look up for the key and return
// This function was not tested in a real application. May contain a bug.
```

```
function Localize(key as string, default as string) as string
Var s as string
```

```
s=app.BundleLocalizedStringMBS(key)
if s="" then
s=default // For Mac OS Classic and Windows
elseif s=key then // returns key for not localized items
msgbox "For developer: the key """+key+"" was not localized for any language."
end if
```

```
return s
end sub
```

```
// use like this
FileOpen.text=Localize("FileOpenLabel", "Open...")
```

Notes: The table parameter is optional to specify which ".strings"-file to use. without table or table="" the "Localizable.strings" file is used by Mac OS X.

Returns "" (empty string) on Mac OS Classic or Windows.

See also:

- 29.1.8 BundleLocalizedStringMBS(key as string) as string

633

29.1.10 BundleResourceFolderItemLocalizedMBS(ResourceName as string, ResourceType as string, SubDirectory as string) as folderitem

Plugin Version: 4.0, Platform: macOS, Targets: Desktop only.

Function: Searches inside the application bundle for a file.

Example:

```
Var f as FolderItem
f=app.BundleResourceFolderItemLocalizedMBS("logo","jpg","")
MsgBox f.NativePath
```

Notes: ResourceName is the filename of the resource file.

ResourceType is the file extension.

SubDirectory is the name of the directory.

This function will take care for localization folders.

29.1.11 BundleResourceFolderMBS as folderitem

Platform: macOS, Targets: Desktop only.

Function: Returns a folderitem to the resource file inside the bundle.

Example:

```
Var f As FolderItem = App.BundleResourceFolderMBS
```

```
If f = Nil Then
MsgBox "Failed?"
Else
MsgBox f.NativePath
End If
```

29.1.12 HideMeMBS as boolean

Plugin Version: 3.2, Platform: macOS, Targets: Desktop only.

Function: Hides the current application.

Notes: Returns true if the current application was hidden.

29.1.13 HideOthersMBS as boolean

Plugin Version: 3.2, Platform: macOS, Targets: Desktop only.

Function: Hides all applications except the current one.

Notes: Returns true if the other applications were hidden.

29.1.14 LaunchTimeMBS as Double

Platform: macOS, Targets: Desktop only.

Function: Returns the the ticks value at launch time.

Example:

```
Var d as date
```

```
d=new date
d.totalSeconds=d.totalSeconds-(ticks-app.launchTimeMBS)/60
```

```
msgbox d.longdate+" "+d.longtime
```

Notes: This value must be converted to the secondsvalue like in the example.

-1 on Windows.

For Mac OS X see the DarwinResourceUsageMBS class.

29.1.15 ProcessTimeMBS as Double

Platform: macOS, Targets: Desktop only.

Function: Returns the current CPU time for the process.

Example:

```
msgbox "I got "+format(app.ProcessTimeMBS/100,"0.00")+ " seconds CPU time till now."
```

Notes: On Mac OS X, the OS counts how much CPU time in given to the current process.

Returns -1 in case of an error.

You can do this function yourself if you like:

```
Var s as SoftDeclareMBS
```

```
s=new SoftDeclareMBS
```

```
if s.LoadLibrary("System.Framework") then
if s.LoadFunction("clock") then
if s.Call(0,nil) then
MsgBox "clock: "+str(s.Result)+chr(13)+"app.ProcessTime: "+str(app.ProcessTime)
end if
end if
end if
```

You may need to add Error Checking code.

29.1.16 Properties

29.1.17 FrontmostMBS as boolean

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns true if the application is frontmost.

Example:

```
app.frontmostmbs=True 'bring app to front
```

Notes: This property is for read and write. Writing to it brings the app to front. Setting this property to false does nothing.

If you need to make some other application frontmost, you can use `app.HideMeMBS` or use the `ProcessMBS` class.

Implemented for Windows in version 19.2. When querying, we check if the front window belongs to our application. When setting the plugin builds a list of all FileMaker windows. Then we activate the top most of them and move all other windows in z order behind.

(Read and Write computed property)

29.2 module CallDelegateCrashSafeMBS

29.2.1 module CallDelegateCrashSafeMBS

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The module to call a delegate and catch a crash.

Example:

Try

```
System.DebugLog "Try the call..."
```

```
CallDelegateCrashSafeMBS AddressOf TestMethod
```

```
System.DebugLog "Call succeeded!"
```

```
Catch c As DelegateCrashExceptionMBS
```

```
System.DebugLog "Caught exception!"
```

```
System.DebugLog c.Message
```

End Try

Notes: We can catch the following conditions:

- Invalid pointer access
- Invalid instruction
- Floating point exception
- Wrong system call
- Broken pipe
- Bus error

We may crash ourselves on stack corruption and we print the stack trace to stderr for macOS and Linux for debugging.

If a crash is caught, there may be memory leaked as objects allocated in the called method are not freed. The call function can not be used recursively and only on one thread at the time.

Blog Entries

- [News from the MBS Xojo Plugins Version 25.0](#)
- [MBS Xojo Plugins, version 24.6pr3](#)

- [Stacktraces for Windows](#)
- [MBS Xojo Plugins, version 23.1pr2](#)
- [News from the MBS Xojo Plugins in Version 23.0](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.0](#)
- [MBS Xojo Plugins, version 23.0pr5](#)
- [Crash Protection for Xojo methods](#)
- [MBS Xojo Plugins, version 22.6pr1](#)

Xojo Developer Magazine

- [21.3, page 10: News](#)
- [21.2, page 9: News](#)

29.2.2 Methods

29.2.3 CallDelegateCrashSafeMBS(m as DelegateCrashSafeMBS)

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Calls a method that may crash.

Example:

Try

```
System.DebugLog "Try the call..."
```

```
CallDelegateCrashSafeMBS AddressOf TestMethod
```

```
System.DebugLog "Call succeeded!"
```

```
Catch c As DelegateCrashExceptionMBS
```

```
System.DebugLog "Caught exception!"
```

```
System.DebugLog c.Message
```

End Try

Notes: Raises DelegateCrashExceptionMBS if a crash happens.

The call function can not be used recursively and only on one thread at the time.

29.2.4 PrintBacktraceMBS

Plugin Version: 25.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Prints the stack trace.

Notes: Utility function to help debugging.

29.2.5 Properties

29.2.6 CallCounter as Integer

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The number of calls made.

Notes: (Read only property)

29.2.7 CrashCounter as Integer

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The number of crashes caught.

Notes: (Read only property)

29.2.8 Delegates

29.2.9 DelegateCrashSafeMBS()

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate definition for a simple method to call.

Blog Entries

- [Stacktraces for Windows](#)

Xojo Developer Magazine

- [21.1, page 9: News](#)

29.3 module CallDelegatesMBS

29.3.1 module CallDelegatesMBS

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The module for the call delegate methods.

Notes: Only supported on Xojo 2013r1 and newer.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [New in MBS Xojo Plugins in version 23.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.1](#)
- [MBS Xojo Plugins, version 23.1pr5](#)
- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.0](#)

29.3.2 Methods

29.3.3 CallDelegateOnMainThreadMBS(m as __delegateMBS)

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a delegate on the main thread.

Example:

CallDelegateOnMainThreadMBS [AddressOf](#) testString, "Hello"

CallDelegateOnMainThreadMBS [AddressOf](#) testNumbers, 5, 6

CallDelegateOnMainThreadMBS [AddressOf](#) TestNoParameters

CallDelegateOnMainThreadMBS [AddressOf](#) Test, "4th call", 4

Notes: We call the delegate on the main thread later with passing a parameter

Our plugin supports various combinations of up to 2 parameters of type string, variant, boolean, double, integer and object.

If you need more, please contact us.

Only supported on Xojo 2013r1 and newer.

If the method you call has optional parameters, you must pass all values or Xojo will throw a compile error.

29.3.4 CallDelegateOnPreemptiveThreadMBS(m as __delegateMBS) as Boolean

Plugin Version: 23.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Calls a delegate on a preemptive thread.

Notes: Experimental function to play with preemptive-threads in Xojo apps.

Some rules:

- No objects, no strings, no arrays
- Use Ptr, not Memoryblocks
- Crashes in debugger, so only use in built app
- Use #Pragma BackgroundTasks False
- Use #Pragma BreakOnExceptions False
- Use #Pragma StackOverflowChecking False
- Use #Pragma NilObjectChecking False
- No exceptions
- Delegate must be stored in global variable to avoid it getting freed early.
- You can pass up to one ptr as data parameter.
- Declares may work, but should be pre-used to make sure they don't raise an exception and are loaded.

uses thread pool on iOS, macOS and Windows. Otherwise regular threads.

See also:

- 29.3.5 CallDelegateOnPreemptiveThreadMBS(m as __delegatePtrMBS, data as Ptr) as Boolean 641

29.3.5 CallDelegateOnPreemptiveThreadMBS(m as __delegatePtrMBS, data as Ptr) as Boolean

Plugin Version: 23.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a delegate on a preemptive thread and passes a ptr.

Notes: Experimental function to play with preemptive-threads in Xojo apps.

Some rules:

- No objects, no strings, no arrays
- Use Ptr, not Memoryblocks
- Crashes in debugger, so only use in built app

- Use `#Pragma BackgroundTasks False`
- Use `#Pragma BreakOnExceptions False`
- Use `#Pragma StackOverflowChecking False`
- Use `#Pragma NilObjectChecking False`
- No exceptions
- Delegate must be stored in global variable to avoid it getting freed early.
- You can pass up to one ptr as data parameter.
- Declares may work, but should be pre-used to make sure they don't raise an exception and are loaded.

uses thread pool on iOS, macOS and Windows. Otherwise regular threads.

See also:

- 29.3.4 `CallDelegateOnPreemptiveThreadMBS(m as __delegateMBS) as Boolean`

29.4 Globals

29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread after the given delay in seconds.

Example:

```
if CallMethodLaterMBS(window1, "Test", 5.0) then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with no parameters and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread after the given delay in seconds.

Returns true on success and false on failure.

The time given is just a roughly suggestions. Actual time on the method call depends on how busy your application is.

See also:

- 29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean 643
- 29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean 644
- 29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 645

29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread after the given delay in seconds.

Example:

```
if CallMethodLaterMBS(window1, "Test", 4.0, "Hello") then
msgbox "OK"
```

```

else
msgbox "Failed"
end if

```

Notes: The method must be declared on the given class for the target object with one variant parameter and no return values.

The method will be called later (Asynchronously) on the main thread after the given delay in seconds. Useful for performing non thread safe stuff like GUI functions on the main thread.

Returns true on success and false on failure.

The time given is just a roughly suggestions. Actual time on the method call depends on how busy your application is.

See also:

- 29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean 643
- 29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean 644
- 29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 645

29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread after the given delay in seconds.

Example:

```

if CallMethodLaterMBS(window1, "Test", 3.0, "Hello", "World") then
msgbox "OK"
else
msgbox "Failed"
end if

```

Notes: The method must be declared on the given class for the target object with two variant parameters and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread after the given delay in seconds.

Returns true on success and false on failure.

The time given is just a roughly suggestions. Actual time on the method call depends on how busy your

application is.

See also:

- 29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean 643
- 29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean 643
- 29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 645

29.4.4 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant, value3 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread after the given delay in seconds.

Example:

```
if CallMethodLaterMBS(window1, "Test", 3.0, "Hello", "World", 5) then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with three variant parameters and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread after the given delay in seconds.

Returns true on success and false on failure.

The time given is just a roughly suggestions. Actual time on the method call depends on how busy your application is.

Blog Entries

- [MBS Xojo Plugins, version 22.1pr1](#)
- [MBS Xojo / Real Studio Plugins, version 15.5pr1](#)

See also:

- 29.4.1 CallMethodLaterMBS(target as object, name as string, afterDelay as Double) as boolean 643
- 29.4.2 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant) as boolean 643

- 29.4.3 CallMethodLaterMBS(target as object, name as string, afterDelay as Double, value1 as Variant, value2 as Variant) as boolean 644

29.4.5 CallMethodMBS(target as object, name as string) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object.

Example:

```
if CallMethodMBS(window1, "Test") then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with no parameters and no return values.

Returns true on success and false on failure.

See also:

- 29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean 646
- 29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean 647
- 29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 647

29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object.

Example:

```
if CallMethodMBS(window1, "Test", "Hello") then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with one variant parameter and no return values.

Returns true on success and false on failure.

See also:

- 29.4.5 CallMethodMBS(target as object, name as string) as boolean 646
- 29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean 647
- 29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 647

29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object.

Example:

```
if CallMethodMBS(window1, "Test", "Hello", "World") then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with two variant parameters and no return values.

Returns true on success and false on failure.

See also:

- 29.4.5 CallMethodMBS(target as object, name as string) as boolean 646
- 29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean 646
- 29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean 647

29.4.8 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object.

Example:

```
if CallMethodMBS(window1, "Test", "Hello", "World", 5) then
msgbox "OK"
```

```

else
msgbox "Failed"
end if

```

Notes: The method must be declared on the given class for the target object with three variant parameters and no return values.

Returns true on success and false on failure.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [About new Call functions](#)
- [MBS REALbasic Plugins, version 10.4pr8](#)

See also:

- 29.4.5 CallMethodMBS(target as object, name as string) as boolean 646
- 29.4.6 CallMethodMBS(target as object, name as string, value1 as Variant) as boolean 646
- 29.4.7 CallMethodMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean 647

29.4.9 CallMethodOnMainThreadMBS(target as object, name as string) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread.

Example:

```

if CallMethodOnMainThreadMBS(window1, "Test") then
msgbox "OK"
else
msgbox "Failed"
end if

```

Notes: The method must be declared on the given class for the target object with no parameters and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread.

Returns true on success and false on failure.

Deprecated, please use CallDelegatesMBS.CallDelegateOnMainThreadMBS instead.

See also:

- 29.4.10 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant) as boolean
649
- 29.4.11 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean
649
- 29.4.12 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean
650

29.4.10 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread.

Example:

```
if CallMethodOnMainThreadMBS(window1, "Test", "Hello") then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with one variant parameter and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread.

Returns true on success and false on failure.

Deprecated, please use CallDelegatesMBS.CallDelegateOnMainThreadMBS instead.

See also:

- 29.4.9 CallMethodOnMainThreadMBS(target as object, name as string) as boolean
648
- 29.4.11 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean
649
- 29.4.12 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant) as boolean
650

29.4.11 CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant) as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread.

Example:

```
if CallMethodOnMainThreadMBS(window1, "Test", "Hello", "World") then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with two variant parameters and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread.

Returns true on success and false on failure.

Deprecated, please use `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead.

See also:

- 29.4.9 `CallMethodOnMainThreadMBS(target as object, name as string)` as boolean 648
- 29.4.10 `CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant)` as boolean 649
- 29.4.12 `CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant)` as boolean 650

29.4.12 `CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant, value3 as Variant)` as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calls a method on the target object on the main thread.

Example:

```
if CallMethodOnMainThreadMBS(window1, "Test", "Hello", "World", 5) then
msgbox "OK"
else
msgbox "Failed"
end if
```

Notes: The method must be declared on the given class for the target object with three variant parameters and no return values.

The method will be called later (Asynchronously) on the main thread. Useful for performing non thread safe stuff like GUI functions on the main thread.

Returns true on success and false on failure.

Deprecated, please use `CallDelegatesMBS.CallDelegateOnMainThreadMBS` instead.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [About new Call functions](#)
- [MBS REALbasic Plugins, version 10.4pr8](#)

See also:

- 29.4.9 `CallMethodOnMainThreadMBS(target as object, name as string)` as boolean 648
- 29.4.10 `CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant)` as boolean 649
- 29.4.11 `CallMethodOnMainThreadMBS(target as object, name as string, value1 as Variant, value2 as Variant)` as boolean 649

29.4.13 `SetThreadNameMBS(name as string)`

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the name of current thread.

Example:

```
SetThreadNameMBS "SQL Query Thread"
```

Notes: This is useful to see thread name in crash reports or sample reports.

Blog Entries

- [MBS Xojo plug-ins in version 16.0](#)
- [Thread Names](#)
- [MBS Xojo / Real Studio Plugins, version 15.5pr2](#)

29.4.14 `CountProcessesMBS as Integer`

Platform: macOS, Targets: Desktop, Console & Web.

Function: Counts how many processes are running.

Notes: Short for this function:

```
function CountProcesses as Integer
  Var n as Integer
  Var p as ProcessMBS

  p=new ProcessMBS
  p.getFirstProcess
  n=1
  while p.getNextProcess
    n=n+1
  wend

  return n
end if
```

29.5 class ConsoleApplication

29.5.1 class ConsoleApplication

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Console only.

Function: Extends the ConsoleApplication class inside Xojo.

29.5.2 Methods

29.5.3 ArgumentsMBS as String()

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Console only.

Function: Queries arguments of process.

Notes: Returns nil in case of any error (low memory).

First entry in result is path to current app.

Other entries are parameters, without any quotes.

Works for Console, Desktop and Web projects. Version 19.4 declares this method for ConsoleApplication and Application class.

29.5.4 LaunchTimeMBS as double

Plugin Version: 19.4, Platform: macOS, Targets: Console only.

Function: Returns the the ticks value at launch time.

Example:

```
Var d as date
```

```
d=new date
d.totalSeconds=d.totalSeconds-(ticks-app.launchTimeMBS)/60
```

```
msgbox d.longdate+" "+d.longtime
```

Notes: This value must be converted to the secondsvalue like in the example.

-1 on Windows.

For Mac OS X see the DarwinResourceUsageMBS class.

Works for Console, Desktop and Web projects. Version 19.4 declares this method for ConsoleApplication and Application class.

29.5.5 ProcessTimeMBS as double

Plugin Version: 19.4, Platform: macOS, Targets: Console only.

Function: Returns the current CPU time for the process.

Example:

```
msgbox "I got "+format(app.ProcessTimeMBS/100,"0.00")+ " seconds CPU time till now."
```

Notes: On Mac OS X, the OS counts how much CPU time in given to the current process.

On Mac OS 9 you can get this information for any process using the ProcessMBS class.

Returns -1 in case of an error.

You can do this function yourself if you like:

```
Var s as SoftDeclareMBS
```

```
s=new SoftDeclareMBS
```

```
if s.LoadLibrary("System.Framework") then
if s.LoadFunction("clock") then
if s.Call(0,nil) then
MsgBox "clock: "+str(s.Result)+chr(13)+"app.ProcessTime: "+str(app.ProcessTime)
end if
end if
end if
```

You may need to add Error Checking code.

Works for Console, Desktop and Web projects. Version 19.4 declares this method for ConsoleApplication and Application class.

29.6 class DelegateCrashExceptionMBS

29.6.1 class DelegateCrashExceptionMBS

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The exception raised when a delegate call crashes.

Notes: Raised in CallDelegateCrashSafeMBS function.

Message defines what caused the crash and ErrorNumber is the signal number (platform dependent!).

Message contains the reason, e.g.

- Delegate crashed with invalid pointer access.
- Delegate crashed with illegal instruction.
- Delegate crashed with floating point error.
- Delegate crashed with wrong system call.
- Delegate crashed with broken pipe.
- Delegate crashed with memory bus error.

Subclass of the RuntimeException class.

Blog Entries

- [News from the MBS Xojo Plugins in Version 23.0](#)
- [Crash Protection for Xojo methods](#)
- [MBS Xojo Plugins, version 22.6pr1](#)

29.7 class DesktopApplication

29.7.1 class DesktopApplication

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Extends the Application class inside Xojo.

29.7.2 Methods

29.7.3 ApplicationNameMBS as string

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns the name of the application.

Example:

```
msgbox "Hi. This application is named "+app.ApplicationNameMBS
```

Notes: This may not be the exact same name than the filename.

29.7.4 ArgumentsMBS as String()

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Queries arguments of process.

Notes: Returns nil in case of any error (low memory).

First entry in result is path to current app.

Other entries are parameters, without any quotes.

Blog Entries

- [Console and GUI in one project](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 15.2](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr1](#)

Xojo Developer Magazine

- [13.5, page 8: News](#)

29.7.5 BundleFolderMBS as folderitem

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns a folderitem to the bundle's mail folder.

Blog Entries

- [MBS Real Studio Plugins, version 13.0pr9](#)

29.7.6 BundleLocalizedStringMBS(key as string) as string

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns the localized string for the given key from the default table.

Notes: A short version for BundleLocalizedString which uses the default table.

See also:

- [29.7.7 BundleLocalizedStringMBS\(key as string,fromtable as string\) as string](#)

657

29.7.7 BundleLocalizedStringMBS(key as string,fromtable as string) as string

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns the localized string for the given key and table.

Example:

```
// You may make yourself a function to look up for the key and return
// This function was not tested in a real application. May contain a bug.
```

```
function Localize(key as string, default as string) as string
  Var s as string
```

```
s=app.BundleLocalizedStringMBS(key)
if s="" then
s=default // For Mac OS Classic and Windows
elseif s=key then // returns key for not localized items
msgbox "For developer: the key """+key+"""" was not localized for any language."
end if
```

```
return s
end sub
```

```
// use like this
FileOpen.text=Localize("FileOpenLabel", "Open...")
```

Notes: The table parameter is optional to specify which ".strings"-file to use. without table or table="" the "Localizable.strings" file is used by Mac OS X.

Returns "" (empty string) on Mac OS Classic or Windows.

See also:

- 29.7.6 BundleLocalizedStringMBS(key as string) as string

29.7.8 BundleResourceFolderItemLocalizedMBS(ResourceName as string, ResourceType as string, SubDirectory as string) as folderitem

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Searches inside the application bundle for a file.

Example:

```
Var f as FolderItem
f=app.BundleResourceFolderItemLocalizedMBS("logo","jpg","")
MsgBox f.NativePath
```

Notes: ResourceName is the filename of the resource file.

ResourceType is the file extension.

SubDirectory is the name of the directory.

This function will take care for localization folders.

29.7.9 BundleResourceFolderMBS as folderitem

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns a folderitem to the resource file inside the bundle.

Example:

```
Var f As FolderItem = App.BundleResourceFolderMBS
```

```
If f = Nil Then
MsgBox "Failed?"
Else
MsgBox f.NativePath
End If
```

Blog Entries

- [MBS Xojo Plugins, version 21.4pr2](#)
- [MBS Real Studio Plugins, version 13.0pr9](#)

29.7.10 HideMeMBS as boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Hides the current application.

Notes: Returns true if the current application was hidden.

29.7.11 HideOthersMBS as boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Hides all applications except the current one.

Notes: Returns true if the other applications were hidden.

29.7.12 LaunchTimeMBS as double

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns the ticks value at launch time.

Example:

```
Var d as date
```

```
d=new date
```

```
d.totalSeconds=d.totalSeconds-(ticks-app.launchTimeMBS)/60
```

```
msgbox d.longdate+" "+d.longtime
```

Notes: This value must be converted to the secondsvalue like in the example.

-1 on Windows.

For Mac OS X see the DarwinResourceUsageMBS class.

29.7.13 ProcessTimeMBS as double

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns the current CPU time for the process.

Example:

```
msgbox "I got "+format(app.ProcessTimeMBS/100,"0.00")+ " seconds CPU time till now."
```

Notes: On Mac OS X, the OS counts how much CPU time is given to the current process.

Returns -1 in case of an error.

29.7.14 Properties

29.7.15 FrontmostMBS as Boolean

Plugin Version: 21.5, Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns true if the application is frontmost.

Example:

```
app.frontmostmbs=True 'bring app to front
```

Notes: This property is for read and write. Writing to it brings the app to front. Setting this property to false does nothing.

If you need to make some other application frontmost, you can use `app.HideMeMBS` or use the `ProcessMBS` class.

Implemented for Windows, too. When querying, we check if the front window belongs to our application. When setting the plugin builds a list of all FileMaker windows. Then we activate the top most of them and move all other windows in z order behind.

(Read and Write computed property)

29.8 class EnvironmentMBS

29.8.1 class EnvironmentMBS

Platform: Windows, Targets: All.

Function: Allows you to access system wide environment properties.

Blog Entries

- [MBS Real Studio Plugins, version 12.1pr3](#)

29.8.2 Methods

29.8.3 Add(name as string,value as string) as boolean

Platform: Windows, Targets: All.

Function: Add a property to the list.

Example:

```
Var e as EnvironmentMBS
```

```
e=new EnvironmentMBS
if e.add("authorname","Christian") then
msgbox "ok"
else
msgbox "fail"
end if
```

Notes: Setting this value to some new value will currently only change this value for your application and all child applications.

29.8.4 Get(name as string) as string

Platform: Windows, Targets: All.

Function: Returns the value for a named property.

Example:

```
Var e as EnvironmentMBS
```

```
e=new EnvironmentMBS
msgbox e.get("windir") // normally c:\windows
```

29.8.5 Lines as string()

Plugin Version: 12.1, Platform: Windows, Targets: All.

Function: Returns array of definition lines of environment variables.

Example:

```
Var e as new EnvironmentMBS
msgbox join(e.lines, endofline)
```

Notes: Returned strings are in format "name=value".

29.8.6 Name(Index as Integer) as string

Platform: Windows, Targets: All.

Function: Returns the name of the environment propertie number n.

29.8.7 Names as string()

Plugin Version: 12.1, Platform: Windows, Targets: All.

Function: Returns array of names of environment variables.

Example:

```
Var e as new EnvironmentMBS
msgbox join(e.names, endofline)
```

29.8.8 Update

Platform: Windows, Targets: All.

Function: Updates the object to the current environment properties.

Notes: This method is called by the constructor and the methods which change the environment properties.

29.8.9 Properties

29.8.10 Count as Integer

Platform: Windows, Targets: All.

Function: Returns the number of environment properties.

Example:

```
Var e as EnvironmentMBS
```

```
e=new EnvironmentMBS  
msgbox str(e.count)
```

Notes: (Read only property)

29.8.11 Value(Index as Integer) as string

Platform: Windows, Targets: All.

Function: Returns the value of the environment property number n.

Notes: Setting this value to some new value will currently only change this value for your application and all child applications.

(Read and Write computed property)

29.9 class MutexMBS

29.9.1 class MutexMBS

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a mutex.

Notes: A mutex is not locked by default.

See also AtomicIntegerMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr3](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr8](#)
- [MonkeyBread Software Releases the MBS Plugins 8.1](#)

29.9.2 Methods

29.9.3 Lock

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Locks the mutex.

Example:

```
Var m as new MutexMBS
```

```
m.Lock  
MsgBox "Got mutex."
```

```
m.Unlock  
MsgBox "Released mutex."
```

Notes: The function returns as soon as it has access to the mutex.

Always use Lock and Unlock in a pair.

29.9.4 TryLock as boolean

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Locks the mutex if possible.

Example:

```
Var m as new MutexMBS
```

```
if m.TryLock then  
MsgBox "Got mutex."
```

```
m.Unlock  
else  
MsgBox "Failed to get mutex."  
end if
```

Notes: Returns true if we got a lock and false if not.

Always use Lock and Unlock in a pair.

29.9.5 Unlock

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Unlocks the mutex.

Example:

```
Var m as new MutexMBS
```

```
m.Lock  
MsgBox "Got mutex."
```

```
m.Unlock  
MsgBox "Released mutex."
```

Notes: Always use Lock and Unlock in a pair.

29.9.6 Properties

29.9.7 Handle as Integer

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal used reference to the native mutex object.

Example:

```
Var m as new MutexMBS  
MsgBox "Handle: "+hex(m.Handle)
```

Notes: Windows Mutex or PThread Mutex.
(Read and Write property)

29.9.8 Tag as Variant

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: An object reference for your use.

Notes: Just for convenience.

This property was added so you can use this property for a reference so you won't need to subclass this class and add custom properties.

(Read and Write property)

29.10 class NamedMutexMBS

29.10.1 class NamedMutexMBS

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a named mutex.

Notes: Using a mutex on Windows and a semaphore on MacOS and Linux.

Raises `UnsupportedOperationException` exceptions if something fails.

See also `MutexMBS` class.

Blog Entries

- [MBS Xojo Plugins, version 24.3pr4](#)
- [New in the MBS Xojo Plugins Version 20.2](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.2](#)
- [MBS Xojo Plugins, version 20.2pr2](#)

29.10.2 Methods

29.10.3 Constructor(Name as string)

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor to create a new mutex.

Notes: Name should be a simple text, but unique, so we recommend reverse domain notation.

The name should not contain a slash.

If mutex exists already, we open it. The mutex lives until the last application closed it.

If name matches the name of an existing event, semaphore, waitable timer, job, or file-mapping object, the function fails.

Raises an `UnsupportedOperationException` on failure.

You can open the same mutex twice and get different Xojo objects from the plugin for both.

29.10.4 Lock

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Locks the mutex.

Notes: Blocks until mutex is locked.

Please call `Unlock` later to unlock.

To avoid you locking up the app, we limit locking to one minute. After that you get a timeout error.

29.10.5 TryLock as boolean

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tries to lock the mutex.

Notes: Returns true if locked or false if the lock can't be set.

If false is returned another app may have a lock currently.

Please call `Unlock` later to unlock if you get back true.

29.10.6 Unlock

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Unlocks the mutex.

29.10.7 Properties

29.10.8 Handle as Integer

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal object reference.

Notes: (Read only property)

29.10.9 Locked as Boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the mutex was locked here.

Notes: We keep track whether you call `Lock/TryLock` here vs. `Unlock` function.

(Read only property)

29.10.10 Name as String

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the mutex.

Notes: (Read only property)

29.10.11 Tag as Variant

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag value.

Notes: You can use it as you like to store whatever value is needed.
(Read and Write property)

29.11 class ProcessMBS

29.11.1 class ProcessMBS

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Adds the class ProcessMBS to Xojo to get details about all running applications.

Example:

```
// Shows how to use the processInformationCFDictionary function to
// decide on Mac OS X whether an application is a background only one.

Var myProcess as ProcessMBS
Var myCFDictionary as CFDictionaryMBS
Var myCFBoolean as CFBooleanMBS
Var myCFObject as CFObjectMBS

myProcess=new ProcessMBS
myProcess.GetCurrentProcess
myCFDictionary = CFDictionaryMBS(myProcess.processInformationCFDictionary)
myCFObject = myCFDictionary.Value(newcfstringmbs("LSBackgroundOnly"))
myCFBoolean=CFBooleanMBS(myCFObject)

if myCFBoolean.Value then
MsgBox "is background"
else
MsgBox "is not background"
end if
```

Notes: Requires Windows 95 or Windows 2000 to run on Windows.

To find all windows on Windows, use the WindowsListMBS class.

To find all windows on Mac OS X, use the CGSWindowListMBS class.

For newer code on Mac, please check NSRunningApplicationMBS class.

See also WindowsProcessDiagnosticInfoMBS class for Windows.

Blog Entries

- [MBS Plugins updated for Xojo 2019r2](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr4](#)
- [Notes on Windows 8](#)
- [MBS Real Studio Plugins, version 12.1pr2](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)

- [MBS REALbasic Plugins, version 10.4pr6](#)
- [MBS REALbasic plug-in 9.6](#)
- [MonkeyBread Software Releases the MBS REALbasic plug-ins 8.6](#)

29.11.2 Methods

29.11.3 Bundle as folderitem

Plugin Version: 4.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the bundle path.

Notes: Maybe nil if not available.

Returns on Mac OS X the path to the main bundle of the process.

29.11.4 BundleID as string

Plugin Version: 7.5, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the bundle identifier.

Notes: Maybe "" if not available.

Returns on Mac OS X the identifier to the main bundle of the process.

29.11.5 CurrentProcessID as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the process ID of the application.

Example:

```
MsgBox str(ProcessMBS.CurrentProcessID)
```

29.11.6 GetCurrentProcess

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Load the current Process' information into the class.

Notes: This should always be your Xojo application.

29.11.7 GetfirstProcess

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Load the first Process into the class.

Example:

```
Var p as ProcessMBS
p=new ProcessMBS
p.GetfirstProcess ' get first
do
msgBox p.name
loop until not p.GetNextProcess ' get next till no more
```

29.11.8 GetFrontProcess

Platform: macOS, Targets: Desktop, Console & Web.

Function: Load the frontmost Process' information into the class.

29.11.9 GetNextProcess as boolean

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Load the next Process into the class.

Example:

```
Var p as ProcessMBS
p=new ProcessMBS
p.GetfirstProcess ' get first
do
msgBox p.name
loop until not p.GetNextProcess ' get next till no more
```

Notes: Returns true if successfull.

29.11.10 GetProcess(serial as memoryblock)

Platform: macOS, Targets: Desktop, Console & Web.

Function: Load the process information for the process with the given serial.

29.11.11 KillProcess as Integer

Plugin Version: 2.7, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Kills a Process on Mac OS X or Windows.

Example:

```
Var p as ProcessMBS
p=new ProcessMBS
p.getcurrentprocess
msgbox str(p.killprocess)
```

Notes: Returns -1 if function is not available. Else returns a Mac OS error code where 0 equals "no error".

Sets the lasterror property.

Lasterror codes on Windows:

- 0 - failed
- 1 - normal exit
- 2 - forced exit

If you kill the main process of a task, this can kill the sub processes, too.

See also:

- 29.11.12 KillProcess(ProcessID as Integer, timeout as Integer) as Integer 673
- 29.11.13 KillProcess(timeoutms as Integer) as Integer 674

29.11.12 KillProcess(ProcessID as Integer, timeout as Integer) as Integer

Plugin Version: 10.3, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Kills a Process on Mac OS X or Windows.

Example:

```
const pid = 12345 // put the PID here
call ProcessMBS.KillProcess pid, 100
```

Notes: Returns -1 if function is not available. Else returns a Mac OS error code where 0 equals "no error".

Returns on Windows:

- 0 - failed
- 1 - normal exit
- 2 - forced exit

The timeout is in milliseconds.

If you kill the main process of a task, this can kill the sub processes, too.

See also:

- 29.11.11 KillProcess as Integer 673
- 29.11.13 KillProcess(timeoutms as Integer) as Integer 674

29.11.13 KillProcess(timeoutms as Integer) as Integer

Plugin Version: 4.2, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Kills a Process on Mac OS X or Windows.

Example:

```
Var p as ProcessMBS
p=new ProcessMBS
p.getcurrentprocess
msgbox str(p.killprocess)
```

Notes: Returns -1 if function is not available. Else returns a Mac OS error code where 0 equals "no error".

Sets the lasterror property.

Lasterror codes on Windows:

0 - failed

1 - normal exit

2 - forced exit

The timeout is in milliseconds. If you don't specify a timeout, 1000 ms are used.

If you kill the main process of a task, this can kill the sub processes, too.

See also:

- 29.11.11 KillProcess as Integer 673
- 29.11.12 KillProcess(ProcessID as Integer, timeout as Integer) as Integer 673

29.11.14 MacProcessSerial as memoryblock

Plugin Version: 2.7, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the serial number for this process as a memoryblock with binary content.

Notes: On Mac OS every process as its own unique ID. (If you launch one application two times, they may have everything equal except the process serial.)

The serial itself is a 8byte long memoryblock (two integer).

Returns nil on any error.

29.11.15 ProcessInformationCFDictionary as object

Plugin Version: 3.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a CFDictionaryMBS object with the properties of the process.

Example:

```

Var p as ProcessMBS
Var d as CFDictionaryMBS

p=new ProcessMBS

p.GetCurrentProcess

d=CFDictionaryMBS(p.ProcessInformationCFDictionary)

if d=nil then
MsgBox "This example works only on Mac OS X 10.2!"
else
// use information in dictionary
end if

```

Notes: Only available on Mac OS X 10.2 or newer.

(And in Mac OS X 10.2 is a bug which may lead into a crash by using this function. Fixed in Mac OS X 10.3)

The function returns a CFDictionaryMBS, but to avoid plugin dependencies, the function is declared as Object, so RB 4.5 don't crash when an application uses the process plugin and doesn't use the CF Plugin.

Result:

An immutable CFDictionary containing these keys and their values. Keys marked with an '*' are optional. Over time more keys may be added.

29.11.16 QuitProcess as Integer

Plugin Version: 3.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Quits a Process on Mac OS.

Example:

Key Name	Type
"PSN"	CFNumber, kCFNumberLongLongType
"Flavor"	CFNumber, kCFNumberSInt32
"Attributes"	CFNumber, kCFNumberSInt32
"ParentPSN"	CFNumber, kCFNumberLongLong
"FileType"	CFString, file type
"FileCreator"	CFString, file creator
"pid"	CFNumber, kCFNumberLongType
"LSBackgroundOnly"	CFBoolean
"LSUIElement"	CFBoolean
"IsHiddenAttr"	CFBoolean
"IsCheckedInAttr"	CFBoolean
"RequiresClassic"	CFBoolean
"RequiresCarbon"	CFBoolean
"LSUserQuitOnly"	CFBoolean
"LSUIPresentationMode"	CFNumber, kCFNumberShortType
"BundlePath"	CFString
kIOBundleExecutableKey	CFString
kIOBundleNameKey	CFString
kIOBundleIdentifierKey	CFString

```

Var p as ProcessMBS
p=new ProcessMBS
p.getcurrentprocess
msgbox str(p.QuitProcess)

```

Notes: Returns -1 if function is not available. Else returns a Mac OS error code where 0 equals "no error". Sets the lasterror property.

29.11.17 SameAs(other as ProcessMBS) as boolean

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns true if two ProcessMBS objects have the same Process.

29.11.18 SetFrontProcessWithOptions(options as Integer)

Plugin Version: 4.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Moves the current process to the front.

Example:

```

Var p as ProcessMBS

```

```
// move my front window to the front
p=new ProcessMBS
p.GetCurrentProcess
p.SetFrontProcessWithOptions 1
```

Notes: If you pass 0 in the options parameter, the process is activated and all process windows are brought forward.

If you pass `kSetFrontProcessFrontWindowOnly` (value 1), the process is activated and the frontmost non-floating window is brought forward. (Mac OS X only)

29.11.19 SetServiceMode(ismode as boolean) as boolean

Plugin Version: 2.7, Platform: Windows, Targets: Desktop, Console & Web.

Function: If an application is a service, it is not listed inside the Windows force quit window.

Notes: Works only on Windows 95, 98 and ME.

29.11.20 TransformProcessType(mode as Integer) as Integer

Plugin Version: 15.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: Changes the 'type' of the process specified in the psn parameter.

Notes: Given a psn for an application, this call transforms that application into the given type. Foreground applications have a menu bar and appear in the Dock. Background applications do not appear in the Dock, do not have a menu bar (and should not have windows or other user interface). UIElement applications do not have a menu bar, do not appear in the dock, but may in limited circumstances present windows and user interface. If a foreground application is frontmost when transformed into a background application, it is first hidden and another application is made frontmost. A UIElement or background-only application which is transformed into a foreground application is not brought to the front (use `SetFrontProcess()` after the transform if this is required) nor will it be shown if it is hidden (even if hidden automatically by being transformed into a background-only application), so the caller should use `ShowHideProcess()` to show the application after it is transformed into a foreground application. Applications can only transform themselves; this call cannot change the type of another application.

Returns error code (0 for success).

29.11.21 TransformToForegroundApplication as Integer

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: Transforms a background application to a foreground application.

Notes: Returns a Mac OS error code. Returns -1 if the function is not available. You need to select a process first with Get*Process functions.

Requires Mac OS X 10.3.

You can use this call to transform a background-only application into a foreground application. A foreground application appears in the Dock (and in the Force Quit dialog) and contains a menu bar. This function does not cause the application to be brought to the front; you must set FrontProcess=true to do so.

While available starting in 10.3 "Panther," calling TransformProcessType on a UIElement application in 10.3 or 10.4 will return paramErr (-50), rendering it only usable on a full BackgroundOnly? application. Leopard supports foregrounding both UIElement and BackgroundOnly? applications.

29.11.22 Update

Platform: macOS, Targets: Desktop, Console & Web.

Function: Updates the classes information.

Notes: Use it if you want the current CPUTime value after some time.

29.11.23 WinFullProcessImagePath as string

Plugin Version: 9.6, Platform: Windows, Targets: Desktop, Console & Web.

Function: Retrieves the full name of the executable image for the specified process.

Notes: Returns empty string on any error like missing permissions.

The path is in the device style "Device\Harddisk0\Partition1\WINNT\System32\test.exe".

Requires Windows Vista or Windows Server 2008 or newer.

29.11.24 WinGetPriorityClass(ProcessID as Integer) as Integer

Plugin Version: 12.1, Platform: Windows, Targets: Desktop, Console & Web.

Function: Retrieves the priority class for the specified process.

Notes: You can pass ProcessID -1 to get the current process.

This value, together with the priority value of each thread of the process, determines each thread's base priority level.

If the function succeeds, the return value is the priority class of the specified process.

The process's priority class is one of the following values:

Name	Value	Description
ABOVE_NORMAL_PRIORITY_CLASS	&h00008000	Process that has priority above NORMAL_PRIORITY_CLASS but below HIGH_PRIORITY_CLASS.
BELOW_NORMAL_PRIORITY_CLASS	&h00004000	Process that has priority above IDLE_PRIORITY_CLASS but below NORMAL_PRIORITY_CLASS.
HIGH_PRIORITY_CLASS	&h00000080	Process that performs time-critical tasks that must be executed immediately for it to run correctly. The threads of a high-priority class process preempt the threads of normal or idle priority class processes. An example is the Task List, which must respond quickly when called by the user, regardless of the load on the operating system. Use extreme care when using the high-priority class, because a high-priority class CPU-bound application can use nearly all available cycles.
IDLE_PRIORITY_CLASS	&h00000040	Process whose threads run only when the system is idle and are preempted by the threads of any process running in a higher priority class. An example is a screen saver. The idle priority class is inherited by child processes.
NORMAL_PRIORITY_CLASS	&h00000020	Process with no special scheduling needs.
REALTIME_PRIORITY_CLASS	&h00000100	Process that has the highest possible priority. The threads of a real-time priority class process preempt the threads of all other processes, including operating system processes performing important tasks. For example, a real-time process that executes for more than a very brief interval can cause disk caches not to flush or cause the mouse to be unresponsive.

Every thread has a base priority level determined by the thread's priority value and the priority class of its process. The operating system uses the base priority level of all executable threads to determine which thread gets the next slice of CPU time. Threads are scheduled in a round-robin fashion at each priority level, and only when there are no executable threads at a higher level will scheduling of threads at a lower level take place.

For a table that shows the base priority levels for each combination of priority class and thread priority value, see Scheduling Priorities.

Priority class is maintained by the executive, so all processes have a priority class that can be queried.

For Mac and Linux, please use Priority.

29.11.25 WinModulePath as string

Plugin Version: 9.6, Platform: Windows, Targets: Desktop, Console & Web.

Function: Returns the path the first code file in this process.

Notes: Returns empty string on any error like missing permissions.

The path is in the typical style "C:\path\name.exe".

29.11.26 WinProcessImagePath as string

Plugin Version: 9.6, Platform: Windows, Targets: Desktop, Console & Web.

Function: Retrieves the name of the executable file for the specified process.

Notes: Returns empty string on any error like missing permissions.
The path is in the device style "Device\Harddisk0\Partition1\WINNT\System32\test.exe".

Requires Windows XP or Windows Server 2003 or newer.

29.11.27 WinSetPriorityClass(ProcessID as Integer, PriorityClass as Integer) as Integer

Plugin Version: 12.1, Platform: Windows, Targets: Desktop, Console & Web.

Function: Sets the priority class for the specified process.

Notes: You can pass ProcessID -1 to get the current process.

This value together with the priority value of each thread of the process determines each thread's base priority level.

The process's priority class is one of the following values:

Name	Value	Description
ABOVE_NORMAL_PRIORITY_CLASS	&h00008000	Process that has priority above NORMAL_PRIORITY_CLASS but below HIGH_PRIORITY_CLASS.
BELOW_NORMAL_PRIORITY_CLASS	&h00004000	Process that has priority above IDLE_PRIORITY_CLASS but below NORMAL_PRIORITY_CLASS.
HIGH_PRIORITY_CLASS	&h00000080	Process that performs time-critical tasks that must be executed immediately for it to run correctly. The threads of a high-priority class process preempt the threads of normal or idle priority class processes. An example is the Task List, which must respond quickly when called by the user, regardless of the load on the operating system. Use extreme care when using the high-priority class, because a high-priority class CPU-bound application can use nearly all available cycles.
IDLE_PRIORITY_CLASS	&h00000040	Process whose threads run only when the system is idle and are preempted by the threads of any process running in a higher priority class. An example is a screen saver. The idle priority class is inherited by child processes.
NORMAL_PRIORITY_CLASS	&h00000020	Process with no special scheduling needs.
REALTIME_PRIORITY_CLASS	&h00000100	Process that has the highest possible priority. The threads of a real-time priority class process preempt the threads of all other processes, including operating system processes performing important tasks. For example, a real-time process that executes for more than a very brief interval can cause disk caches not to flush or cause the mouse to be unresponsive.

Every thread has a base priority level determined by the thread's priority value and the priority class of its process. The operating system uses the base priority level of all executable threads to determine which thread gets the next slice of CPU time. Threads are scheduled in a round-robin fashion at each priority level, and only when there are no executable threads at a higher level will scheduling of threads at a lower level take place.

For a table that shows the base priority levels for each combination of priority class and thread priority value, see Scheduling Priorities.

Priority class is maintained by the executive, so all processes have a priority class that can be queried.

Every thread has a base priority level determined by the thread's priority value and the priority class of its process. The system uses the base priority level of all executable threads to determine which thread gets the next slice of CPU time. The `SetThreadPriority` function enables setting the base priority level of a thread relative to the priority class of its process. For more information, see [Scheduling Priorities](#).

The `*_PRIORITY_CLASS` values affect the CPU scheduling priority of the process. For processes that perform background work such as file I/O, network I/O, or data processing, it is not sufficient to adjust the CPU scheduling priority; even an idle CPU priority process can easily interfere with system responsiveness when it uses the disk and memory. Processes that perform background work should use the `PROCESS_MODE_BACKGROUND_BEGIN` and `PROCESS_MODE_BACKGROUND_END` values to adjust their resource scheduling priorities; processes that interact with the user should not use `PROCESS_MODE_BACKGROUND_BEGIN`.

If a process is in background processing mode, the new threads it creates will also be in background processing mode. When a thread is in background processing mode, it should minimize sharing resources such as critical sections, heaps, and handles with other threads in the process, otherwise priority inversions can occur. If there are threads executing at high priority, a thread in background processing mode may not be scheduled promptly, but it will never be starved.

Each thread can enter background processing mode independently using `SetThreadPriority`. Do not call `SetPriorityClass` to enter background processing mode after a thread in the process has called `SetThreadPriority` to enter background processing mode. After a process ends background processing mode, it resets all threads in the process; however, it is not possible for the process to know which threads were already in background processing mode.

On any error the value is zero.

For Mac and Linux, please use `Priority`.

29.11.28 Properties

29.11.29 CPUTime as Integer

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the number of the ticks this Process did get to calculate.

Notes: Unit is 1/60 of a second.

On Mac OS X this value is not available and normally 0.

(Read only property)

29.11.30 CurrentProcess as boolean

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns true if the application is in the current Process.

Notes: This should always be true because you are running Xojo as the current ProcessMBS while you ask for this.

(Read only property)

29.11.31 flags as Integer

Plugin Version: 4.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The flags for this process on Mac OS.

Example:

```

Sub Open()
// List all visible processes

Var s as string
Var p as ProcessMBS
Var d As CFDictionaryMBS
Var LSBackgroundOnly, LSUIElement As CFStringMBS
Var b As CFBooleanMBS
Var obj As CFObjectMBS
Var visible as Boolean

const modeOnlyBackground=&h00000400

LSUIElement=NewCFStringMBS("LSUIElement")
LSBackgroundOnly = NewCFStringMBS("LSBackgroundOnly")

p = new ProcessMBS
p.getFirstProcess

do
visible=True

// Mac OS X
d = CFDictionaryMBS(p.ProcessInformationCFDictionary)

if d <>nil Then
obj = d.Value(LSBackgroundOnly)
if obj isa CFBooleanMBS then
b=CFBooleanMBS(obj)

if b.value Then
visible=false
end
end if

obj= d.Value(LSUIElement)
if obj isa CFBooleanMBS then
b=CFBooleanMBS(obj)

```

```

if b.value Then
visible=false
end
end if
end if

// Mac OS 9

if BitwiseAnd(p.Flags,modeOnlyBackground)<>0 then
visible=False
end if

if visible then
ListBox1.AddRow p.Name
end if

loop until not p.GetNextProcess
End Sub

```

Notes: Returns 0 on any error.

Process mode flags. These flags indicate whether the process is an application or desk accessory. For applications, this field also returns information specified in the application's 'SIZE' resource. This information is returned as flags. You can refer to these flags by using these constants:

modeDeskAccessory	&h00020000
modeMultiLaunch	&h00010000
modeNeedSuspendResume	&h00004000
modeCanBackground	&h00001000
modeDoesActivateOnFGSwitch	&h00000800
modeOnlyBackground	&h00000400
modeGetFrontClicks	&h00000200
modeGetAppDiedMsg	&h00000100
mode32BitCompatible	&h00000080
modeHighLevelEventAware	&h00000040
modeLocalAndRemoteHLEvents	&h00000020
modeStationeryAware	&h00000010
modeUseTextEditServices	&h00000008

(Read only property)

29.11.32 FrontProcess as boolean

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns true if the application is in the front of all others.

Example:

```
Var p as new ProcessMBS
p.GetCurrentProcess
p.FrontProcess=true 'bring app to front
```

Notes: This property is for read and write. Writing to it brings the Process to front.
(Read and Write property)

29.11.33 lasterror as Integer

Plugin Version: 3.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The last error code.

Notes: As this property is introduced with MBS Plugin 3.1, only a few functions set it.
0 means no error, -1 means not available and anything else is a Mac OS error code.
(Read and Write property)

29.11.34 LaunchProcess as ProcessMBS

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the application which launched the current one.

Notes: For some applications like the "loginwindow" it is nil, because there was no app to launch it other than the system itself.
(Read only property)

29.11.35 MacCreator as string

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the Mac Creator code for the application.

Notes: (Read only property)

29.11.36 MacType as string

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the Mac Type code for the application.

Notes: Possible values:

"APPL" normal application
"appe" background application (Classic)
"APPC" Control panel application (Classic)
"APPD" Desktop Accessory (Classic)
"FNDR" Finder

others may be possible.
(Read only property)

29.11.37 MemoryFree as Integer

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the size of the free space in the Process' memory partition.

Notes: On Mac OS X normally 0.

(Read only property)

29.11.38 MemorySize as Integer

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns size of the Process' memory partition.

Notes: On Mac OS X normally 0.

(Read only property)

29.11.39 Name as string

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the name of this application.

Notes: Added Windows support in MBS Plugin 2.7 for this property.

(Read only property)

29.11.40 Path as folderitem

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the path of the Process' executable file.

Notes: Maybe nil if not available.

Use this folderitem to get an icon for the Process.

(and this icon may be nil!)

(Read only property)

29.11.41 ProcessID as Integer

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the PID of the process for use in unix command line tools.

Example:

Function LoginWindowPID() *as Integer*

// returns Process ID of Loginwindow on Mac OS X.

// or 0 on error

Var p *as* ProcessMBS

p=new ProcessMBS

p.GetFirstProcess

do

if p.Name="loginwindow" **then**

Return p.ProcessID

end if

loop until not p.GetNextProcess

End Function

Notes: Works only on Mac OS X.

See the "CPU Focus" application for an example.

Note that this call does not make sense for Classic applications, since they all share a single UNIX process ID.

(Read only property)

29.11.42 Visible as boolean

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns true if the application is visible.

Example:

// hide iTunes

```

Var p as new ProcessMBS

p.GetFirstProcess
do
if p.BundleID="com.apple.iTunes" then
p.Visible=false
exit
end if
loop until not p.GetNextProcess

```

Notes: Works only on Mac OS X.
Returns false if not supported.

Sets the lasterror property.
(Read and Write property)

29.11.43 Priority as Integer

Plugin Version: 10.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: The priority of the process.

Example:

```

Var p as new ProcessMBS

p.GetCurrentProcess

MsgBox str(p.priority) // shows 0

p.priority = 5

MsgBox str(p.priority) // shows 5

```

Notes: 20 (least) to -20 (max). Setting the value may be limited due to permissions.
(Read and Write computed property)
See also:

- 29.11.44 Priority(ProcessID as Integer) as Integer

29.11.44 Priority(ProcessID as Integer) as Integer

Plugin Version: 12.1, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Sets or gets priority for a process on Mac and Linux.

Notes: Value is in the range -20 to 20. The default priority is 0; lower priorities cause more favorable scheduling.

For Windows, please use `WinSetPriorityClass` and `WinGetPriorityClass`.

(Read and Write computed property)

See also:

- 29.11.43 Priority as Integer

687

29.11.45 Constants

Process Transformation Types

Constant	Value	Description
<code>kProcessTransformToBackgroundApplication</code>	2	Turn application in a background application. functional in Mac OS X 10.7 and later
<code>kProcessTransformToForegroundApplication</code>	1	Turn application in a foreground application.
<code>kProcessTransformToUIElementApplication</code>	4	Turn application in a user interface element application. functional in Mac OS X 10.7 and later

Chapter 30

RAMStream

30.1 class RAMStreamMBS

30.1.1 class RAMStreamMBS

Platforms: macOS, Linux, Windows, Targets: All.

Function: Allows you to use a memory bases stream.

Notes: If you need to concat lot's of streams, just make a RAMstream and write to it.

In the example folder is a project called "test ramstream speed" which tests the speed on concating 10000 strings which each 1000 bytes in length.

On my G4 Dual 1000 Mhz, I get this result:

String only: 22063 ticks

RAMStream: 18 ticks

Blog Entries

- [MBS Real Studio Plugins, version 12.5pr2](#)

30.1.2 Methods

30.1.3 close

Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

Notes: There is no need to call this method except you want to free all resources of this object now without waiting for Xojo to do it for you.

30.1.4 Constructor(InitialSize as Integer=0)

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new RAMStream.

Example:

```
Var r as RAMStreamMBS
r=new RAMStreamMBS(1000000)
r.write "Hello"
```

Notes: To avoid memory fragmentation the memory grows in 32 KByte chunks.

The parameter you give is the size for the first allocation. So if you only need 2 KByte, you just pass 2048. If you don't know the size, you can pass 0 or a negative number to get the default initial size which is 32 KByte.

You can change the GrowSize property to use a different growing size. Performance is better if resizing the memory buffer is minimized.

The stream can grow to around 2 GB.
On low memory, the initial resize will fail and length will be 0.

30.1.5 Look(count as Integer) as string

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes, but does not move the current position.

Example:

```
Var r as RamStreamMBS // your stream
Var s as string
s=r.look(100)
```

Notes: Like the lookahead property in a socket.

30.1.6 LookBlock(count as Integer) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes into a memoryblock, but leaves the current position untouched.

Example:

```
Var b as RAMStreamMBS
Var s as memoryblock
?...
s=b.LookBlock(5)
```

30.1.7 LookByte as Integer

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a byte, but leaves the current position untouched.

30.1.8 LookLong as Integer

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads an integer, but leaves the current position untouched.

30.1.9 LookShort as Integer

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a short, but leaves the current position untouched.

30.1.10 Read(count as Integer) as string

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes into a string.

Example:

```
Var b as RAMStreamMBS
Var s as string
?...
s=b.read(5)
```

30.1.11 ReadBlock(count as Integer) as memoryblock

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes into a memoryblock.

Example:

```
Var b as RAMStreamMBS
Var s as memoryblock
?...
s=b.ReadBlock(5)
```

30.1.12 Readbyte as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads an 8bit Byte from the stream.

Example:

```
Var b as RAMStreamMBS
Var i as Integer
?...
i=B.readbyte
```

30.1.13 ReadLong as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a signed 32bit Integer from the stream.

Example:

```
Var b as RAMStreamMBS
Var i as Integer
?...
i=B.readlong
```

Notes: This function is affected by the LittleEndian Setting.

30.1.14 ReadShort as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a signed 16bit Integer from the stream.

Example:

```
Var b as RAMStreamMBS
Var i as Integer
?...
i=B.readshort
```

Notes: This function is affected by the LittleEndian Setting.

30.1.15 Write(data as string)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes bytes from a string to file.

30.1.16 WriteBlock(data as memoryblock,count as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes count bytes from a memoryblock to file.

Example:

```
Var b as RAMStreamMBS
Var m as memoryblock
?...
b.writeblock m,m.size
```

30.1.17 WriteByte(data as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes a byte to file.

30.1.18 WriteLong(data as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes an 32bit integer to file.

Notes: This method is affected by the LittleEndian Setting.

30.1.19 WriteShort(data as Integer)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes an 16bit integer to file.

Notes: This method is affected by the LittleEndian Setting.

30.1.20 Properties

30.1.21 EOF as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if you are at the end of the stream.

Notes: (Read only property)

30.1.22 GrowSize as Integer

Plugin Version: 7.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The size in bytes used to grow the ramstream backbuffer.

Notes: Default is 32 Kilobytes.

If the value is bigger than zero, it will be used as the allocation increase. Else the default value is used.
(Read and Write property)

30.1.23 Length as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the current length of the stream.

Notes: You can truncate the stream by setting this property.
(Read and Write property)

30.1.24 LittleEndian as boolean

Platforms: macOS, Linux, Windows, Targets: All.

Function: Decides whether to convert integers read or wrote to the file.

Notes: See Xojos binarystream for more details.

For native platform you may set "littleendian=targetwin32".

(Read and Write property)

30.1.25 MemoryUsed as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: How much memory is currently used for this Stream.

Notes: (Read and Write property)

30.1.26 Position as Integer

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns current position.

Notes: You can set the current file stream position using this property.

(Read and Write property)

30.2 Globals

30.2.1 CreateRamStreamMBS(InitialSize as Integer = 0) as RamStreamMBS

Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new RAMStream.

Example:

```
Var r as RAMStreamMBS
r=CreateRamStreamMBS(1000000)
r.write "Hello"
```

Notes: To avoid memory fragmentation the memory grows in 32 KByte chunks.

The parameter you give is the size for the first allocation. So if you only need 2 KByte, you just pass 2048. If you don't know the size, you can pass 0 or a negative number to get the default initial size which is 32 KByte.

You can change the `GrowSize` property to use a different growing size. Performance is better if resizing the memory buffer is minimized.

The stream can grow to around 2 GB.
Returns nil on low memory.

Chapter 31

Registration

31.1 class RegistrationEngineMBS

31.1.1 class RegistrationEngineMBS

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: An engine to add serial number generation to your application.

Example:

```
Var r as new RegistrationEngineMBS
```

```
r.Field(0)="Hello World"  
r.Field(1)="MyProduct 2008"
```

```
MsgBox r.Calc // shows: ICYR-RES4-UXQ1
```

Notes: If you want to add serial numbers to your application, you can use this class to write yourself a serial number generator application.

You can later in your application use this class to verify the serial number.

You may want to design your application that the serial number is written to your preferences file so you can verify it every time your application is launched. But if you do this, also save some computer related value there so you can avoid people cloning the system. For example the MAC ID, the user name, the systemfolder creation date. This way you detect whenever a preferences file is copied to another machine.

For generate and later verify serial numbers the setup of the class must match. So carefully check how your serial number should look like. A format like "MA-1234-5678-9012-V1" can be good. Use a prefix and suffix so make serial numbers from different products and different major versions look different. 12 letters are normally enough for a serial number. Especially if you use alpha numeric characters. You can of course

change the alphabet string and use lower case letters.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.4](#)
- [MBS Xojo Plugins, version 17.4pr4](#)
- [Handllng Sales and Serial Numbers](#)
- [MBS Plugins 11.1 Release notes](#)
- [MBS Real Studio Plugins, version 11.1pr11](#)
- [MBS REALbasic plug-in 9.6](#)
- [MBS REALbasic plug-ins version 9.5](#)

31.1.2 Methods

31.1.3 Calc as string

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the serial number.

31.1.4 Verify(s as string) as boolean

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Verifies the serial number.

Example:

```
Var r As New RegistrationEngineMBS
```

```
r.Field(0)="Hello World"
r.Field(1)="MyProduct 2008"
```

```
Var key As String = r.Calc
```

```
If r.Verify(key) Then
  MsgBox "OK"
Else
  MsgBox "Failed"
End If
```

Notes: If you only use upper case letter for your serial numbers, use a function like uppercase before calling verify. This way you avoid that a serial number "abc" is invalid because it is not "ABC".

Returns true on success and false on failure.

31.1.5 Properties

31.1.6 Alphabet as String

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The characters to use for serial numbers.

Example:

```
Var r as new RegistrationEngineMBS
```

```
r.Field(0)="Hello World"  
r.Field(1)="MyProduct 2008"
```

```
r.Alphabet="ABC123"  
MsgBox r.Calc // shows: A1AC-2312-12CC
```

```
r.Alphabet="0123456789"  
MsgBox r.Calc // shows: 4308-2905-4013
```

Notes: Must have at least two characters.

Characters must be unique.

Default is "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ".

(Read and Write property)

31.1.7 BlockLength as Integer

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Length of the blocks in the serial number.

Example:

```
Var r as new RegistrationEngineMBS
```

```
r.Field(0)="Hello World"  
r.Field(1)="MyProduct 2008"
```

```
r.NumberLength=10  
r.BlockLength=5  
MsgBox r.Calc // shows: ICYRR-ES4UX
```

```
r.NumberLength=16  
r.BlockLength=4
```

MsgBox r.Calc // shows: ICYR-RES4-UXQ1

Notes: If this value is 0, no delimiters are used.
(Read and Write property)

31.1.8 Delimiter as String

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The delimiter string to use for separating blocks in the serial number.

Notes: Default is "-".

(Read and Write property)

31.1.9 Mode as Integer

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The mode of operation.

Notes: Currently always 0.

Set to 1 to use 128bit instead of 64bit number.

(Read and Write property)

31.1.10 NumberLength as Integer

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The maximum length of the number.

Example:

```
Var r as new RegistrationEngineMBS
```

```
r.Field(0)="Hello World"
```

```
r.Field(1)="MyProduct 2008"
```

```
r.NumberLength=10
```

```
r.BlockLength=5
```

```
MsgBox r.Calc // shows: ICYRR-ES4UX
```

```
r.NumberLength=16
```

```
r.BlockLength=4
```

```
MsgBox r.Calc // shows: ICYR-RES4-UXQ1
```

Notes: (Read and Write property)

31.1.11 Platform as Integer

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: All.

Function: Which platform to use for platform specific keys.

Notes: Value is set in the constructor to current platform, but you can change it if you want to run the key generator on a different platform.

Values are:

- 1 Mac
- 2 Windows
- 3 Linux

All other values behave like PlatformSpecificKeys is false.

This value is only used if PlatformSpecificKeys is true.

(Read and Write property)

31.1.12 PlatformSpecificKeys as Boolean

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether you want platform specific keys.

Notes: Set to false if you want to have Windows, Mac OS X and Linux all use the same serial numbers. Set to true if you want to different serial numbers depending on the platform.

Default is false.

(Read and Write property)

31.1.13 Prefix as String

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The prefix for the serial number string.

Notes: (Read and Write property)

31.1.14 Seed as Integer

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: A seed value you can set to get your serial number more unique.

Notes: (Read and Write property)

31.1.15 Suffix as String

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The suffix for the serial number string.

Example:

```
Var r as new RegistrationEngineMBS
```

```
r.Field(0)="Hello World"
r.Field(1)="Xojo"
r.suffix="-RB600CPFUSA-MAC"
```

```
r.BlockLength=8
r.NumberLength=32
```

```
MsgBox r.Calc
```

```
// shows: EV6GG9P2-DI533EV6-GG9P2DI5-33EV6GG9-RB600CPFUSA-MAC
// looks like a Xojo serial number ;-)
```

Notes: (Read and Write property)

31.1.16 Field(index as Integer) as string

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The data fields you can use for personalized serial numbers.

Notes: Fill this fields with all the values you want to use for personalized serial numbers. You should use the name of the user, maybe the postal address. You can use hardware values like the MAC address or software values like the system folder creation date.

Also you may have serial numbers different between version 1.x and 2.x of your application. So you pass version information like "1" or "2". Do not pass values like "1.2.3" as this serial number won't work on "1.2.4".

Index is from 0 to 7.

(Read and Write computed property)

Chapter 32

Resolution

32.1 class DisplayMBS

32.1.1 class DisplayMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Defines a class for accessing the displays of the screen.

Blog Entries

- [MBS Xojo Plugins, version 17.3pr3](#)
- [MBS Xojo Plugins, version 17.2pr6](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr11](#)
- [MBS REALbasic Plugins, version 10.4pr4](#)
- [MBS Plugins 10.3 Release Notes](#)
- [MBS REALbasic plug-ins version 9.5](#)

Xojo Developer Magazine

- [17.5, page 43: What's New in the MBS Plugins](#), With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes

32.1.2 Methods

32.1.3 CanDepth(depth as Integer) As Boolean

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns true if the monitor supports the given color depth at the current ResolutionMBS.

Notes: none.

32.1.4 FadeGamma(intensity as Integer, col As Color)

Platforms: macOS, Windows, Targets: Desktop only.

Function: Sets the gamma of the DisplayMBS

Notes: Intensity has to be between 0 and 1000. 1000 is normal, when it is 0, the whole DisplayMBS is in the color GammaColor.

Direct color display modes do not use color lookup tables and are usually 16, 24, or 32 bit. Not all direct color video boards support loadable gamma ramps.

See the NativeGamma documentation for more details on Windows Gamma details.

32.1.5 FadeGammaTo(intensity as Integer, col As Color, ticks as Integer)

Platforms: macOS, Windows, Targets: Desktop only.

Function: Fades the intensity and color of the gamma slowly in the given number of milliseconds.

Notes: Intensity has to be between 0 and 1000. 1000 is normal, when it is 0, the whole display is in the color GammaColor.

Direct color display modes do not use color lookup tables and are usually 16, 24, or 32 bit. Not all direct color video boards support loadable gamma ramps.

See the NativeGamma documentation for more details on Windows Gamma details.

32.1.6 GetBestResolution(width as Integer, height as Integer, depth as Integer, safe As Boolean) As ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Searches for the best matching resolution.

Notes: Returns the best resolution which size is not smaller than the dimensions given, which has the given depth and, if safe is true, which is safe. If safe is not true, be sure that the frequency works, if you want to switch to it (for example by asking the user).

On Mac OS X till now no resolution can be called safe.

See also:

- 32.1.7 GetBestResolution(width as Integer, height as Integer, safe As Boolean) As ResolutionMBS 705

32.1.7 GetBestResolution(width as Integer, height as Integer, safe As Boolean) As ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Searches for the best matching resolution.

Notes: Returns the best resolution which size is not smaller that the dimensions given, which has the highest depth and, if safe is true, which is safe. If safe is not true, be sure that the frequency works, if you want to switch to it (for example by asking the user).

On Mac OS X till now no resolution can be called safe.

See also:

- 32.1.6 GetBestResolution(width as Integer, height as Integer, depth as Integer, safe As Boolean) As ResolutionMBS 704

32.1.8 GetCurrentResolution As ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns the current resolution.

Notes: Returns the resolution the display is currently set to. This is useful if you want to switch back to the old resolution after you switched to another resolution.

32.1.9 GetLargestResolution(depth as Integer, safe As Boolean) As ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Searches for the largest resolution with the given depth.

Notes: The same as GetBestResolution(width, height, depth, safe), the only difference is that it does not search for the smallest resolution but for the biggest one.

On Mac OS X till now no resolution can be called safe.

See also:

- 32.1.10 GetLargestResolution(safe As Boolean) As ResolutionMBS 705

32.1.10 GetLargestResolution(safe As Boolean) As ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Searches for the largest resolution.

Notes: If safe is true, this method returns only safe resolutions.

On Mac OS X till now no resolution can be called safe.

See also:

- 32.1.9 GetLargestResolution(depth as Integer, safe As Boolean) As ResolutionMBS 705

32.1.11 GetResolution(num as Integer) As ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns a resolution.

Notes: Num has to be between 0 and ResolutionCount(safe)-1.

32.1.12 ResolutionCount(depth as Integer, safe As Boolean) as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns the number of resolutions for a given depth.

Notes: This method returns the number of resolutions that have the given depth and, if safe is true, which are safe.

On Mac OS X till now no ResolutionMBS can be called safe.

See also:

- 32.1.13 ResolutionCount(safe As Boolean) as Integer 706

32.1.13 ResolutionCount(safe As Boolean) as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns the number of resolutions.

Notes: This method returns the total number of resolutions this monitor supports and, if safe is true, which are safe. Safe resolutions are resolutions which will always work.

On Mac OS X till now no ResolutionMBS can be called safe.

See also:

- 32.1.12 ResolutionCount(depth as Integer, safe As Boolean) as Integer 706

32.1.14 SetDepth(depth as Integer) As Boolean

Platforms: macOS, Windows, Targets: Desktop only.

Function: Switches the color depth to the depth given. Returns true if attempt was successful.

Notes: none.

32.1.15 SwitchTo(width as Integer, height as Integer, depth as Integer, safe As Boolean) As Boolean

Platforms: macOS, Windows, Targets: Desktop only.

Function: Switches to the given ResolutionMBS if possible.

Notes: This method tries to switch directly to the best ResolutionMBS which size is bigger or the same as the given dimensions and which depth is the same as the given parameter. It returns true if it was successful. Try this function first with safe set to true. If that fails, try again with false, but after that show a dialog asking the user if that ResolutionMBS is okay. If the user presses escape or clicks cancel, you should immediately switch back to the old ResolutionMBS.

On Mac OS X till now no ResolutionMBS can be called safe.

32.1.16 Update

Plugin Version: 3.1, Platforms: macOS, Windows, Targets: Desktop only.

Function: Updates the properties in this class.

Notes: Call it whenever you detect a screen resolution change to update the content of this class.

In general you should not keep DisplayMBS classes around too long because the display described may go away (one a Powerbook the external display).

32.1.17 Properties

32.1.18 Depth as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Depth of the DisplayMBS.

Notes: Can be 8, 16 or 32. Other screen depths are not well supported.

(Read only property)

32.1.19 displaynum as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: DisplayNum is the number of the DisplayMBS

Notes: The number is always between 0 and DisplayCount-1, where 0 is the main monitor.

(Read only property)

32.1.20 GammaColor as color

Platform: macOS, Targets: Desktop only.

Function: The color for fading the DisplayMBS.

Notes: This property is read-only. To change it, use the methods of the DisplayMBS class and the Switch method of the ResolutionMBS class.

There is no fading for Windows.

(Read only property)

32.1.21 GammaIntensity as Integer

Platform: macOS, Targets: Desktop only.

Function: GammaIntensity is the intensity of the monitor.

Notes: 1000 is normal, if it is 0, the whole DisplayMBS is in the color GammaColor.

This property is read-only. To change it, use the methods of the DisplayMBS class and the Switch method of the ResolutionMBS class.

There is no fading for Windows.

(Read only property)

32.1.22 Height as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Height of the monitor.

Notes: (Read only property)

32.1.23 hz as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: frequency in Hz of that DisplayMBS

Example:

```
Var d as DisplayMBS
```

```
d=GetDisplayMBS(0)
```

```
msgBox str(d.Hz)
```

Notes: On Windows:

Sometimes Hz is 0, this is when Windows itself does not know the frequency (at least it is so in my emulation,

Virtual PC 3.0 with Win95).

The value should never be greater as the real frequency, so you can just ignore this.

(Read only property)

32.1.24 Left as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: The relative left offset to the other monitors.

Notes: Should be 0 for the main monitor.

(Read only property)

32.1.25 Top as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: The relative top offset to the other monitors.

Notes: Should be 0 for the main monitor.

(Read only property)

32.1.26 Width as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Width of this screen.

Notes: (Read only property)

32.1.27 NativeGamma as memoryblock

Plugin Version: 3.0, Platforms: macOS, Windows, Targets: Desktop only.

Function: The native gamma values.

Example:

```
Var d as DisplayMBS // your display
d.nativegamma=newmemoryblock(3072) // all black
```

Notes: 3072 bytes big is this array a 256 RGB values saved in 3 single properties.

Some notes for SetDeviceGammaRamp, which is the Windows function used by the plugin:

SetDeviceGammaRamp doesn't allow all gamma ramps. It checks the gamma ramp; if it is too complex, such as the red flash when the player is shot in Quake, it rejects it.

SetDeviceGammaRamp will not currently make use of a gamma calibrator. This may change in future versions of Image Color Management (ICM), but for Windows 2000, only the DirectDraw API supports the gamma calibrators.

The existing gamma entry points is already used by GetDeviceGammaRamp and SetDeviceGammaRamp. Therefore, the display driver doesn't need to do anything special to support this new interface, as long as it already supports the Win32 Get/SetDeviceGammaRamp functions.

In addition to getting and setting gamma ramps, the new DirectDraw interface allows the new gamma ramp to be calibrated—if a gamma calibrator is installed. The mechanism that DirectDraw uses to register and communicate with the gamma calibrator is an interim mechanism that will be changed in future releases.

DirectDraw looks for an installed software calibrator and passes the gamma value to the software calibrator; the software calibrator in turn adjusts the gamma ramp according to the measured response of the monitor. The calibrator passes the gamma ramp back to DirectDraw, which passes it to the SetDeviceGammaRamp device driver interface (DDI). The result is that the game looks as intended.

For DirectDraw to use the gamma calibrator, the calibrator must register itself with DirectDraw using a registry key; DirectDraw will call it if the application wants the gamma ramp to be calibrated.

In the future, both the method by which DirectDraw communicates with the gamma calibrator, through the DDI, and the method gamma calibrators use to register themselves in the registry will change. But every part of the DirectDraw API is permanent.

ICM is the color management system in Windows; all system-level color management should be handled by ICM. For this reason, downloadable gamma ramp support will be rolled into ICM in the future, making current solutions for gamma calibrators obsolete. Until such time, we have provided a method by which applications can take advantage of the installed base of software calibrators and graphics adapters that support downloadable gamma ramps.

(Read and Write computed property)

32.2 Globals

32.2.1 DisplayCountMBS as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: DisplayCount returns the number of displays connected.

Example:

```
msgbox "You have "+str(DisplayCountMBS)+" screens."
```

32.2.2 GetDisplayMBS(num as Integer) As DisplayMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Returns an object of class DisplayMBS for the given DisplayMBS.

Example:

```
Var d as DisplayMBS
```

```
d=GetDisplayMBS(0)
msgBox str(d.width)+" x "+str(d.height)+" @ "+str(pow(2,d.depth))+" colors"
```

Notes: num must be: $0 \leq \text{num} < \text{displaycount}$

32.2.3 ResolutionLibraryPresentMBS as boolean

Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns true if the DisplayManager library file was found.

Example:

```
if not ResolutionLibraryPresentMBS then
msgbox "I'm missing the shared library 'DisplayManager'"
end if
```

Notes: This library should be present on all Mac OS Classic versions. (It is in the system suitcase.) Returns always true on Windows.

Blog Entries

- [MBS Xojo Plugins, version 20.6pr4](#)

32.2.4 UpdateDisplayCountMBS

Plugin Version: 8.0, Platforms: macOS, Windows, Targets: Desktop only.

Function: Rebuilds the display list.

Notes: If new displays are attached or removed, this method needs to be called.

32.3 class ResolutionMBS

32.3.1 class ResolutionMBS

Platforms: macOS, Windows, Targets: Desktop only.

Function: Defines a class for accessing the resolutions a screen.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr4](#)

32.3.2 Methods

32.3.3 Switch As Boolean

Platforms: macOS, Windows, Targets: Desktop only.

Function: This function switches the DisplayMBS to the ResolutionMBS and returns true if the attempt was successful.

Notes: If IsSafe is false, show a dialog after the switch asking the user if that ResolutionMBS is okay. If the user presses escape or clicks cancel, you should immediately switch back to the old ResolutionMBS. Especially on VGA-Monitors not all listed frequencies work.

On Mac OS X till now no ResolutionMBS can be called safe.

32.3.4 Properties

32.3.5 Depth as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Depth of the ResolutionMBS.

Notes: Can be 8, 16 or 32. Other screen depths are not well supported.
(Read only property)

32.3.6 displaynum as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: DisplayNum is the number of the DisplayMBS

Notes: The number is always between 0 and DisplayCount-1, where 0 is the main monitor.
(Read only property)

32.3.7 Height as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Height of the ResolutionMBS.

Notes: (Read only property)

32.3.8 hz as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: frequency in Hz of that ResolutionMBS

Notes: Sometimes Hz is 0, this is when Windows itself does not know the frequency (at least it is so in my emulation, Virtual PC 3.0 with Win95).
(Read only property)

32.3.9 issafe as boolean

Platforms: macOS, Windows, Targets: Desktop only.

Function: Is it a safe ResolutionMBS?

Notes: On Mac OS X till now no ResolutionMBS can be called safe.
(Read only property)

32.3.10 Left as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: The relative left offset to that ResolutionMBS.

Notes: Should be 0 for the main monitor.
(Read only property)

32.3.11 ResolutionNum as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: DisplayNum is the number of the DisplayMBS.

Notes: ResolutionNum is the number of the DisplayMBS which always is between 0 and GetDisplayMBS(DisplayNum).ResolutionCount(false)-1.

On Windows: `dis.GetCurrentResolution.resolutionNum` is sometimes also -1.
(Read only property)

32.3.12 Top as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: The relative top offset to the other monitors.

Notes: Should be 0 for the main monitor.
(Read only property)

32.3.13 Width as Integer

Platforms: macOS, Windows, Targets: Desktop only.

Function: Width of this ResolutionMBS.

Notes: (Read only property)

Chapter 33

SerialPort

33.1 class SerialPortMBS

33.1.1 class SerialPortMBS

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enumerate serial ports and send/receive data.

Notes: While we compiled this class also for iOS, we don't expect anyone to find a serial port on an iPhone.

Blog Entries

- [MBS Xojo Plugins, version 25.1pr1](#)
- [News from the MBS Xojo Plugins Version 23.4](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.4](#)
- [MBS Xojo Plugins, version 23.4pr6](#)
- [News from the MBS Xojo Plugins in Version 23.0](#)
- [MBS Xojo Plugins, version 22.6pr1](#)
- [News from the MBS Xojo Plugins Version 22.4](#)
- [MBS Xojo Plugins in version 22.4](#)
- [News from the MBS Xojo Plugins Version 22.3](#)
- [MBS Xojo Plugins in version 22.3](#)

Xojo Developer Magazine

- [21.6, page 9: News](#)
- [20.6, page 9: News](#)

33.1.2 Methods

33.1.3 AvailableBytes as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of bytes available in buffer.

33.1.4 Clear

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears incoming buffer.

33.1.5 Close

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Closes serial port.

Notes: This happens automatically in the destructor.

33.1.6 Constructor

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Notes: Please call `Open()` to open a device.

33.1.7 HasDataAvailable as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether data is available to read.

33.1.8 HasLine as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if a line is available.

Notes: The plugin checks the input buffer for a newline character.

33.1.9 List(Mode as Integer = 0) as String()

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns list of names of serial ports.

Example:

```
Var names() as string = MySerialPort.List

for each n as string in names
  PopupDevice.AddRow n
next
```

Notes: Returns list of names of serial ports.

On Windows can list ports with textual name.

Mode can be 1 on Windows to return friendly names of ports.

33.1.10 Open(Name as String, Index as Integer = 0)

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens the serial port.

Example:

```
Var port as new SerialPortMBS

// open a port
port.Open("COM1")

// check the path
MessageBox port.Path
```

Notes: You get the names from List function.

The port is opened with default settings.

Even if port is not showing in list, it may still open here, e.g. virtual COM ports on Windows.

You can open several ports with different connections, e.g. by calling Open once for each SerialPortMBS object.

33.1.11 OpenPath(Path as String)

Plugin Version: 23.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens the special file/device/pipe.

Notes: This allows you to e.g. open a pipe on Windows.

33.1.12 Poll

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Polls to see if there is new data.

Notes: The plugin regularly checks if there is new data and raises the DataAvailable event if needed.

33.1.13 Read(MaxByteSizeToRead as Integer) as String

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads bytes from serial port and returns them as text.

Notes: Reads bytes from serial port and returns them as text.

The text returned may be shorter than length bytes.

See also AvailableBytes function.

33.1.14 ReadByte as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the byte value or an error message.

Notes: This is very useful for reading control characters which you can't easily put in a text in FileMaker. Like 13 for a LF (line feed) or 10 for CR (carriage return).

See also AvailableBytes function.

33.1.15 ReadLine as String

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reads a line of text from serial port.

Notes: The plugin buffers input and reads text until first newline character. Returns text without that newline character.

33.1.16 Resume

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Resumes serial port.

Notes: While suspended, we do not trigger event and clear incoming buffer regularly.

33.1.17 Suspend

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Suspends serial port.

Notes: While suspended, we do not trigger event and clear incoming buffer regularly.

Please call Resume later.

33.1.18 Write(value as String) as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of bytes written or an error message.

Notes: Be aware that text encoding conversion may result in different bytes than expected, so please verify on other side the text encoding matches the expectations.

33.1.19 WriteByte(value as Integer) as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of bytes written.

Notes: This is very useful for sending control characters which you can't easily put in a text in Xojo. Like 13 for a LF (line feed) or 10 for CR (carriage return).

33.1.20 Properties

33.1.21 Handle as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal object reference.

Notes: (Read only property)

33.1.22 Lasterror as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error code.

Notes: The error code is zero for no error.

On Windows you get the windows system error codes and on macOS and Linux the unix system error codes.

The methods and properties raise `IOException` in case of errors and include the error codes.
(Read only property)

33.1.23 Path as String

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The path used to open the serial port.

Example:

```
Var port as new SerialPortMBS
```

```
// open a port  
port.Open("COM1")
```

```
// check the path  
MessageBox port.Path
```

Notes: Set by `Open()` method.

Value is e.g. "COM1"

(Read only property)

33.1.24 Suspended as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks suspension state.

Notes: Checks suspension state.

While suspended, we do not trigger event and clear incoming buffer regularly.

(Read only property)

33.1.25 Tag as Variant

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag property.

Notes: You can store whatever you like in this property to read it later.
(Read and Write property)

33.1.26 BaudRate as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets the baud rate.

Notes: (Read and Write computed property)

33.1.27 CTS as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The current CTS mode.

Notes: If enabled, the CTS (clear-to-send) signal is monitored for output flow control. Than if CTS is turned off, output is suspended until CTS is sent again.
(Read and Write computed property)

33.1.28 DataBits as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets the data bit size.

Notes: Value is 5, 6, 7 or 8.

(Read and Write computed property)

33.1.29 DSR as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets whether DSR mode is used.

Notes: If enabled DSR (data-set-ready) signal is monitored for output flow control.
(Read and Write computed property)

33.1.30 DTR as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets whether DTR (data-terminal-ready) flow control is used.

Notes: (Read and Write computed property)

33.1.31 Parity as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets parity setting.

Notes: Can be 0 (off), 1 (odd) or 2 (even), see constants.

(Read and Write computed property)

33.1.32 RTS as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets whether to use RTS (request-to-send) flow control.

Notes: (Read and Write computed property)

33.1.33 StopBits as Double

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets number of stop bits.

Notes: Can be 1 or 2 on Mac/Linux and 1, 1.5 or 2 on Windows.

(Read and Write computed property)

33.1.34 WinRTSControl as Integer

Plugin Version: 22.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The RTS (request-to-send) flow control.

Notes: This property gives you direct access to the RTS control setting on Windows.

And this is set to either disabled or handshake by the RTS property.

Raises an exception if used on macOS or Linux.

(Read and Write computed property)

33.1.35 XON as Boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries or sets if XON/XOFF flow control is used.

Notes: (Read and Write computed property)

33.1.36 Events**33.1.37 DataAvailable**

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event called when data is available.

Notes: Please call the Read methods here to process data.

33.1.38 Constants

Parity

Constant	Value	Description
kParityEven	2	Even parity
kParityNone	0	No parity
kParityOdd	1	Odd parity

RTS Flow Control Settings

Constant	Value	Description
kWinRTSControlDisabled	0	Disables the RTS line when the device is opened and leaves it disabled.
kWinRTSControlEnabled	1	Enables the RTS line when the device is opened and leaves it on.
kWinRTSControlHandshake	2	Enables RTS handshaking. The driver raises the RTS line when the "type-ahead" (input) buffer is less than one-half full and lowers the RTS line when the buffer is more than three-quarters full.
kWinRTSControlToggle	3	Specifies that the RTS line will be high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line will be low.

Chapter 34

Shell

34.1 Globals

34.1.1 ConsoleExecuteMBS(path as folderitem, arguments() as string, environment() as string) as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Executes a new application.

Notes: Use WindowsShellExecuteMBS on Windows.

Launch GUI tools on Mac OS X using /bin/open.

Returned value is the PID of the new process.

If the execution fails you still get a PID, but this process is terminated in a few milliseconds.

arguments must have at least one member.

See also WindowsProcessMBS and WindowsShellExecuteMBS (Windows only), NSTask (Mac only) and ShellMBS (cross platform).

See also:

- 34.1.2 ConsoleExecuteMBS(path as string, arguments() as string, environment() as string) as Integer
725

34.1.2 ConsoleExecuteMBS(path as string, arguments() as string, environment() as string) as Integer

Plugin Version: 7.7, Platforms: macOS, Linux, Targets: Desktop, Console & Web.

Function: Executes a new application.

Example:

```
// launch an app using open

const sShellPath="/usr/bin/open"

Var a(1) as string
Var e(-1) as string

a(0)="open" // must be the application name
a(1)="/Applications/TextEdit.app" // first parameter

print str(ConsoleExecuteMBS(sShellPath,a,e))

// Launch app binary directly:

Var aa(1) as string
Var ee(-1) as string

Var f as FolderItem

f=GetFolderItem("test.app").Child("Contents").Child("MacOS").Child("test")

aa(0)=f.ShellPath

print str(ConsoleExecuteMBS(f.ShellPath,aa,ee))
```

Notes: Use `WindowsShellExecuteMBS` on Windows.

Launch GUI tools on Mac OS X using `/bin/open`.

Returned value is the PID of the new process.

If the execution fails you still get a PID, but this process is terminated in a few milliseconds.

arguments must have at least one member.

See also `WindowsProcessMBS`, `WindowsShellExecuteAsAdminMBS` and `WindowsShellExecuteMBS` (Windows only), `NSTask` (Mac only) and `ShellMBS` (cross platform).

Blog Entries

- [MBS Xojo Plugins, version 21.5pr2](#)

See also:

- 34.1.1 `ConsoleExecuteMBS(path as folderitem, arguments() as string, environment() as string) as Integer` 725

34.2 class ShellMBS

34.2.1 class ShellMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The class in MBS Plugins to run executables.

Example:

```
Var s As New ShellMBS
```

```
s.Execute "/usr/bin/whoami"
```

```
s.Wait 5
```

```
Var output As String = s.ReadOutput
```

```
Var errors As String = s.ReadError
```

```
Break // see in debugger
```

```
MsgBox "whoami: "+output
```

Notes: Via input stream, we can send data to the process.

And via output and error streams, we can read data coming back.

While this class does the same as Shell class in Xojo, we have the possibility to add more in the future.

For now you can benefit from an alternative implementation with stderr and stdout as two streams. Also we feature terminate methods to kill the child process.

See also `WindowsProcessMBS`, `WindowsShellExecuteAsAdminMBS` and `WindowsShellExecuteMBS` (Windows only), `ConsoleExecuteMBS` and `NSTaskMBS` (Mac only).

Blog Entries

- [News from the MBS Xojo Plugins Version 25.0](#)
- [MBS Xojo Plugins, version 25.0pr7](#)
- [MBS Xojo Plugins, version 23.5pr6](#)
- [News from the MBS Xojo Plugins in Version 23.0](#)
- [MBS Xojo Plugins, version 23.0pr8](#)
- [MBS Xojo Plugins, version 22.6pr1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr6](#)
- [MBS Xojo Plugins, version 20.1pr1](#)

- [MBS Xojo Plugins in version 19.2](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [19.4, page 10: News](#)

34.2.2 Methods

34.2.3 Arguments as String()

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Queries arguments array.

Example:

```
// query execution policy in PowerShell
Var s As New ShellMBS

s.Arguments.Append "-Command"
s.Arguments.Append "Get-ExecutionPolicy -List"

s.Execute "powershell.exe"
s.Wait 5

Var output As String = s.ReadOutput
Var errors As String = s.ReadError

Break // see in debugger
```

Notes: You can add arguments to this array.

34.2.4 CloseInput

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Closes stdin to the process.

Notes: Sometimes this is needed as some processes wait till stdin is closed.

34.2.5 CloseStreams

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Closes the streams.

Notes: Automatically done in destructor.

34.2.6 Constructor

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The constructor.

34.2.7 Destructor

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The destructor.

34.2.8 Execute(Executable as String)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Runs the executable.

Example:

```
// run PowerShell and get help  
Var s As New ShellMBS
```

```
s.SetArguments array("/?")  
s.Execute "powershell.exe"  
s.Wait 5
```

```
Var output As String = s.ReadOutput
```

```
Var errors As String = s.ReadError
```

```
Break // see in debugger
```

Notes: Pass path to executable to run.

Opens the streams stdin, stdout and stderr.

34.2.9 ExitCode as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Queries the exit code of the terminated process.

34.2.10 Poll

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Polls for events.

Notes: This will check for pending event and fire events if needed.

Internally we call this with a timer for the shell object, but you can call this in a tight loop.

34.2.11 ReadError as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reads data from stderr.

Notes: Queries available bytes to return all the data in the buffer.

34.2.12 ReadOutput as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reads data from stdout.

Notes: Queries available bytes to return all the data in the buffer.

34.2.13 SetArguments(arguments()) as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Assigns a new array of arguments.

Example:

```
Var sh as new ShellMBS
```

```
Var args() As String
```

```
args.Append "-c"
```

```
args.Append "5"
```

```
args.Append "monkeybreadsoftware.com"
```

sh.SetArguments args

34.2.14 Terminate(WinExitCode as Integer = 255) as boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Terminates the app with the given exit code.

Notes: ExitCode is only used on Windows.

Returns true on success.

34.2.15 Wait(TimeOut as double = 30.0)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Waits for process to quit.

Notes: Yields time and waits until time passed or process terminates.

34.2.16 WriteInput(data as string) as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Sends data to stdin of the running process.

Notes: Returns number of bytes written.

34.2.17 Properties

34.2.18 ApplicationName as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The path to the application.

Notes: Can be empty when application is part of command line.

Only for Windows.

(Read and Write property)

34.2.19 AvailableBytesError as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Queries number of available bytes on stderr.

Notes: (Read only property)

34.2.20 AvailableBytesOutput as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Queries number of available bytes on stdout.

Notes: (Read only property)

34.2.21 CurrentDirectory as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The current directory for the process.

Notes: You can set this before calling Execute to specify the start directory.

If this parameter is "", the new process will have the same current drive and directory as the calling process.

(Read and Write property)

34.2.22 Domain as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The domain name.

Notes: Only used on Windows.

You can set username and password to run app with different user. Domain is optional to specify network domain.

(Read and Write property)

34.2.23 Environment as Dictionary

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The environment variables to use.

Notes: (Read and Write property)

34.2.24 ErrorCode as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Last native error code.

Notes: Zero is success.

(Read only property)

34.2.25 Executable as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The command line to run.

Notes: On Mac and Linux just the path to executable.

If applicationName is set for Windows, this should only provide parameters.

(Read only property)

34.2.26 IsRunning as Boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Checks whether process is running.

Notes: Returns true if running or false if not.

The process may not terminate if there is still output data in the pipes. So please read error/stdout regularly.

(Read only property)

34.2.27 MergeErrorAndOutput as Boolean

Plugin Version: 25.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether to use one pipe or two pipes for output and error.

Notes: By default this is false and we receive error and output in two different pipes.

Optionally you can merge both to output pipe.

(Read and Write property)

34.2.28 Password as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The password.

Notes: You can set username and password to run app with different user. Domain is optional to specify network domain.

Only for Windows.

(Read and Write property)

34.2.29 PID as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The process ID of the child process.

Notes: (Read only property)

34.2.30 ProcessHandle as Integer

Plugin Version: 19.5, Platform: Windows, Targets: Desktop, Console & Web.

Function: For Windows the process handle.

Notes: (Read only property)

34.2.31 Tag as Variant

Plugin Version: 23.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The tag property.

Notes: You can store whatever you like in this property to read it later.

(Read and Write property)

34.2.32 ThreadHandle as Integer

Plugin Version: 19.5, Platform: Windows, Targets: Desktop, Console & Web.

Function: For Windows the thread handle for the child process.

Notes: (Read only property)

34.2.33 Username as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The user name.

Notes: You can set username and password to run app with different user. Domain is optional to specify network domain.

Only for Windows.

(Read and Write property)

34.2.34 Events

34.2.35 Completed

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: The event called when process terminated.

34.2.36 DataAvailable(AvailableBytesError as Integer, AvailableBytesOutput as Integer)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Event called when new data is available on stderr or stdout.

Notes: You need to use read commands to query data.

Chapter 35

Sort

35.1 module SortMBS

35.1.1 module SortMBS

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Useful sort methods for Xojo arrays.

Example:

```
// fill two arrays
Var test1() As String
Var test2() As String

For i As Integer = 1 To 100000
test1.append Str(i)
test2.append Str(i)
Next

// now sort using Xojo and MBS
Var m1 As Double = Microseconds
test1.Sort
Var m2 As Double = Microseconds
sortArrayMBS test2
Var m3 As Double = Microseconds

// check times
Var d1 As Double = m2-m1
Var d2 As Double = m3-m2

MessageBox d1.ToString+" in Xojo, "+d2.ToString+" in MBS"
```

Notes: Functions to sum up values in arrays, copying values or comparing values is much faster inside the plugin, than if you code this yourself in Xojo. Methods are available as method in SortMBS module as well as using extends on the Array directly.

Allows the usage of delegates for deciding how to sort yourself.

Blog Entries

- [MBS Xojo Plugins, version 25.2pr6](#)
- [MBS Xojo Plugins, version 25.2pr4](#)
- [News from the MBS Xojo Plugins in Version 25.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 25.1](#)
- [MBS Xojo Plugins, version 25.1pr4](#)
- [The SortMBS module](#)
- [MBS Xojo Plugins, version 25.1pr1](#)
- [News from the MBS Xojo Plugins Version 24.2](#)
- [News from the MBS Xojo Plugins Version 24.0](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.0](#)

35.1.2 Methods

35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.8 AddArrayMBS(array1() as Double, array2() as Double) 741
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744
- 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64) 745

35.1.4 AddArrayMBS(array1() as Color, array2() as Color)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant) 745

35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.4 AddArrayMBS(array1() as Color, array2() as Color) 739
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.8 AddArrayMBS(array1() as Double, array2() as Double) 741
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744

35.1.6 AddArrayMBS(array1() as Date, array2() as Date)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.8 AddArrayMBS(array1() as Double, array2() as Double) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744
- 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant) 745

35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.4 AddArrayMBS(array1() as Color, array2() as Color) 739
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744

35.1.8 AddArrayMBS(array1() as Double, array2() as Double)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.4 AddArrayMBS(array1() as Color, array2() as Color) 739
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744

35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.4 AddArrayMBS(array1() as Color, array2() as Color) 739
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64) 745

35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

Example:

```
Var n() As Integer = Array(1,2)
Var a() As Integer = Array(3,4)
```

```
AddArrayMBS n, a
```

```
Break // check n in debugger
```

See also:

- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.8 AddArrayMBS(array1() as Double, array2() as Double) 741
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64) 745
- 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant) 745

35.1.11 AddArrayMBS(array1() as Object, array2() as Object)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.4 AddArrayMBS(array1() as Color, array2() as Color) 739
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741

35.1. <i>MODULE SORTMBS</i>	743
• 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr)	743
• 35.1.14 AddArrayMBS(array1() as String, array2() as String)	744
• 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32)	744
• 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64)	745
• 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant)	745

35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean)	738
• 35.1.4 AddArrayMBS(array1() as Color, array2() as Color)	739
• 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency)	739
• 35.1.6 AddArrayMBS(array1() as Date, array2() as Date)	740
• 35.1.8 AddArrayMBS(array1() as Double, array2() as Double)	741
• 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64)	742
• 35.1.14 AddArrayMBS(array1() as String, array2() as String)	744
• 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32)	744
• 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64)	745
• 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant)	745

35.1.13 AddArrayMBS(array1() as Single, array2() as Single)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean)	738
• 35.1.4 AddArrayMBS(array1() as Color, array2() as Color)	739
• 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency)	739
• 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime)	740

- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744
- 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64) 745
- 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant) 745

35.1.14 AddArrayMBS(array1() as String, array2() as String)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740
- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.8 AddArrayMBS(array1() as Double, array2() as Double) 741
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32) 744

35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean) 738
- 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency) 739
- 35.1.6 AddArrayMBS(array1() as Date, array2() as Date) 740

35.1. MODULE SORTMBS	745
• 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime)	740
• 35.1.8 AddArrayMBS(array1() as Double, array2() as Double)	741
• 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32)	741
• 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64)	742
• 35.1.13 AddArrayMBS(array1() as Single, array2() as Single)	743
• 35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64)	745
• 35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant)	745

35.1.16 AddArrayMBS(array1() as UInt64, array2() as UInt64)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean)	738
• 35.1.4 AddArrayMBS(array1() as Color, array2() as Color)	739
• 35.1.5 AddArrayMBS(array1() as Currency, array2() as Currency)	739
• 35.1.6 AddArrayMBS(array1() as Date, array2() as Date)	740
• 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime)	740
• 35.1.8 AddArrayMBS(array1() as Double, array2() as Double)	741
• 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64)	742
• 35.1.11 AddArrayMBS(array1() as Object, array2() as Object)	742
• 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr)	743
• 35.1.15 AddArrayMBS(array1() as UInt32, array2() as UInt32)	744

35.1.17 AddArrayMBS(array1() as Variant, array2() as Variant)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.3 AddArrayMBS(array1() as Boolean, array2() as Boolean)	738
• 35.1.6 AddArrayMBS(array1() as Date, array2() as Date)	740

- 35.1.7 AddArrayMBS(array1() as DateTime, array2() as DateTime) 740
- 35.1.8 AddArrayMBS(array1() as Double, array2() as Double) 741
- 35.1.9 AddArrayMBS(array1() as Int32, array2() as Int32) 741
- 35.1.10 AddArrayMBS(array1() as Int64, array2() as Int64) 742
- 35.1.11 AddArrayMBS(array1() as Object, array2() as Object) 742
- 35.1.12 AddArrayMBS(array1() as Ptr, array2() as Ptr) 743
- 35.1.13 AddArrayMBS(array1() as Single, array2() as Single) 743
- 35.1.14 AddArrayMBS(array1() as String, array2() as String) 744

35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32) 749
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753

35.1.19 AddMBS(extends array1() as Color, array2() as Color)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747

35.1. MODULE SORTMBS	747
• 35.1.21 AddMBS(extends array1() as Date, array2() as Date)	747
• 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime)	748
• 35.1.23 AddMBS(extends array1() as Double, array2() as Double)	748
• 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32)	749
• 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64)	749
• 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr)	751
• 35.1.28 AddMBS(extends array1() as Single, array2() as Single)	751
• 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64)	753
• 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant)	753

35.1.20 AddMBS(extends array1() as Currency, array2() as Currency)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean)	746
• 35.1.19 AddMBS(extends array1() as Color, array2() as Color)	746
• 35.1.21 AddMBS(extends array1() as Date, array2() as Date)	747
• 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime)	748
• 35.1.23 AddMBS(extends array1() as Double, array2() as Double)	748
• 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32)	749
• 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64)	749
• 35.1.26 AddMBS(extends array1() as Object, array2() as Object)	750
• 35.1.28 AddMBS(extends array1() as Single, array2() as Single)	751
• 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant)	753

35.1.21 AddMBS(extends array1() as Date, array2() as Date)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean)	746
--	-----

- 35.1.19 AddMBS(extends array1() as Color, array2() as Color) 746
- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753
- 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant) 753

35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.21 AddMBS(extends array1() as Date, array2() as Date) 747
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32) 752
- 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant) 753

35.1.23 AddMBS(extends array1() as Double, array2() as Double)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean) 746

35.1. MODULE SORTMBS	749
• 35.1.19 AddMBS(extends array1() as Color, array2() as Color)	746
• 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency)	747
• 35.1.21 AddMBS(extends array1() as Date, array2() as Date)	747
• 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32)	749
• 35.1.26 AddMBS(extends array1() as Object, array2() as Object)	750
• 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr)	751
• 35.1.29 AddMBS(extends array1() as String, array2() as String)	752
• 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32)	752
• 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64)	753

35.1.24 AddMBS(extends array1() as Int32, array2() as Int32)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

• 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean)	746
• 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency)	747
• 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime)	748
• 35.1.23 AddMBS(extends array1() as Double, array2() as Double)	748
• 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64)	749
• 35.1.26 AddMBS(extends array1() as Object, array2() as Object)	750
• 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr)	751
• 35.1.29 AddMBS(extends array1() as String, array2() as String)	752
• 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32)	752
• 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64)	753

35.1.25 AddMBS(extends array1() as Int64, array2() as Int64)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

Example:

```
Var n() As Integer = Array(1,2)
Var a() As Integer = Array(3,4)
```

```
n.AddMBS(a)
```

```
Break // check n in debugger
```

See also:

- 35.1.19 AddMBS(extends array1() as Color, array2() as Color) 746
- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753
- 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant) 753

35.1.26 AddMBS(extends array1() as Object, array2() as Object)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.19 AddMBS(extends array1() as Color, array2() as Color) 746
- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.21 AddMBS(extends array1() as Date, array2() as Date) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753

35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean) 746
- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32) 749
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753

35.1.28 AddMBS(extends array1() as Single, array2() as Single)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean) 746
- 35.1.21 AddMBS(extends array1() as Date, array2() as Date) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32) 749
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753
- 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant) 753

35.1.29 AddMBS(extends array1() as String, array2() as String)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.20 AddMBS(extends array1() as Currency, array2() as Currency) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753
- 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant) 753

35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean) 746
- 35.1.19 AddMBS(extends array1() as Color, array2() as Color) 746
- 35.1.21 AddMBS(extends array1() as Date, array2() as Date) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753

35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.18 AddMBS(extends array1() as Boolean, array2() as Boolean) 746
- 35.1.19 AddMBS(extends array1() as Color, array2() as Color) 746
- 35.1.21 AddMBS(extends array1() as Date, array2() as Date) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.30 AddMBS(extends array1() as UInt32, array2() as UInt32) 752
- 35.1.32 AddMBS(extends array1() as Variant, array2() as Variant) 753

35.1.32 AddMBS(extends array1() as Variant, array2() as Variant)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds the content of the second array to the first one.

See also:

- 35.1.21 AddMBS(extends array1() as Date, array2() as Date) 747
- 35.1.22 AddMBS(extends array1() as DateTime, array2() as DateTime) 748
- 35.1.23 AddMBS(extends array1() as Double, array2() as Double) 748
- 35.1.24 AddMBS(extends array1() as Int32, array2() as Int32) 749
- 35.1.25 AddMBS(extends array1() as Int64, array2() as Int64) 749
- 35.1.26 AddMBS(extends array1() as Object, array2() as Object) 750
- 35.1.27 AddMBS(extends array1() as Ptr, array2() as Ptr) 751
- 35.1.28 AddMBS(extends array1() as Single, array2() as Single) 751
- 35.1.29 AddMBS(extends array1() as String, array2() as String) 752
- 35.1.31 AddMBS(extends array1() as UInt64, array2() as UInt64) 753

35.1.33 CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 756
- 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.35 `CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 755
- 35.1.36 `CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 756
- 35.1.37 `CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 757
- 35.1.38 `CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 758
- 35.1.39 `CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 759
- 35.1.41 `CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 761
- 35.1.43 `CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763
- 35.1.44 `CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 764
- 35.1.45 `CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 766
- 35.1.46 `CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767

35.1.35 `CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.34 `CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.36 `CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 756
- 35.1.38 `CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 758
- 35.1.39 `CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 759
- 35.1.40 `CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 760
- 35.1.41 `CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 761
- 35.1.42 `CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763
- 35.1.43 `CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763
- 35.1.45 `CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 766
- 35.1.46 `CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767

35.1.36 `CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)`

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Copies the content of one array to another array.

Notes: Does nothing if `sourceCount` is 0 or source array is empty.

`sourceCount` can be -2 to use array count - `sourceIndex`.

`destIndex` lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 `CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754

- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 755
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 755

- 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 756
- 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760

- 35.1.41 CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 761
- 35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.44 CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 764
- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766

35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: nDoes nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.^{one}

See also:

- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 756
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.44 CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 764

- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Example:

```
Var s() As Integer
Var d() As Integer
```

```
Redim s(100)
```

```
For i As Integer = 1 To 100
s(i) = i
Next
```

```
Redim d(s.ubound)
```

```
// copy all starting 20th and put at 10th in destination
CopyArrayMBS s, d, 20, -2, 10
```

```
// copy 10 entries starting 30th and put at 40th in destination
CopyArrayMBS s, d, 30, 10, 40
```

```
// copy all
CopyArrayMBS s, d
```

```
Break
```

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

35.1. MODULE SORTMBS

761

- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 755
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- 35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.44 CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 764
- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.41 CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Example:

```
Var s() As Pair
```

```
Var d() As Pair
```

```
Redim s(100)
```

```
For i As Integer = 1 To 100
```

```
s(i) = New pair(i,i)
```

```
Next
```

```
Redim d(s.ubound)
```

```
// copy all starting 20th and put at 10th in destination
```

```
CopyArrayMBS s, d, 20, -2, 10
```

```
// copy 10 entries starting 30th and put at 40th in destination
CopyArrayMBS s, d, 30, 10, 40
```

```
// copy all
CopyArrayMBS s, d
```

```
Break
```

Notes: Does nothing if sourceCount is 0 or source array is empty.
 SourceCount can be -2 to use array count - sourceIndex.
 destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 `CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.35 `CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 755
- 35.1.36 `CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 756
- 35.1.38 `CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 758
- 35.1.40 `CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 760
- 35.1.42 `CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763
- 35.1.43 `CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763
- 35.1.44 `CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 764
- 35.1.46 `CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767
- 35.1.47 `CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767

35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 755
- 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 756
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- 35.1.39 CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 759
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 `CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.34 `CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.35 `CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 755
- 35.1.37 `CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 757
- 35.1.39 `CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 759
- 35.1.40 `CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 760
- 35.1.41 `CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 761
- 35.1.42 `CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763
- 35.1.46 `CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767
- 35.1.47 `CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767

35.1.44 `CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Example:

`Var s() As String`

`Var d() As String`

```

Redim s(100)

For i As Integer = 1 To 100
s(i) = i.ToString
Next

Redim d(s.ubound)

// copy all starting 20th and put at 10th in destination
CopyArrayMBS s, d, 20, -2, 10

// copy 10 entries starting 30th and put at 40th in destination
CopyArrayMBS s, d, 30, 10, 40

// copy all
CopyArrayMBS s, d

Break

```

Notes: Does nothing if sourceCount is 0 or source array is empty.
SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 `CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.34 `CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.37 `CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 757
- 35.1.38 `CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 758
- 35.1.39 `CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 759
- 35.1.40 `CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 760
- 35.1.41 `CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 761
- 35.1.42 `CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 763

- 35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 766
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.45 CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.34 CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 755
- 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 756
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.41 CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 761
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.44 CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 764
- 35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767
- 35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 767

35.1.46 CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 754
- 35.1.35 CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 755
- 35.1.36 CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 756
- 35.1.37 CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 757
- 35.1.38 CopyArrayMBS(source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 758
- 35.1.40 CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 760
- 35.1.41 CopyArrayMBS(source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 761
- 35.1.42 CopyArrayMBS(source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.43 CopyArrayMBS(source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 763
- 35.1.44 CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 764

35.1.47 CopyArrayMBS(source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.33 `CopyArrayMBS(source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.34 `CopyArrayMBS(source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 754
- 35.1.35 `CopyArrayMBS(source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 755
- 35.1.36 `CopyArrayMBS(source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 756
- 35.1.37 `CopyArrayMBS(source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 757
- 35.1.39 `CopyArrayMBS(source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 759
- 35.1.40 `CopyArrayMBS(source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 760
- 35.1.44 `CopyArrayMBS(source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 764
- 35.1.45 `CopyArrayMBS(source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 766
- 35.1.46 `CopyArrayMBS(source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 767

35.1.48 `CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)`

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.51 `CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 771
- 35.1.52 `CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 772
- 35.1.53 `CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 773
- 35.1.54 `CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 774
- 35.1.55 `CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 774
- 35.1.58 `CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 777
- 35.1.59 `CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 778
- 35.1.60 `CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 779
- 35.1.61 `CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 780
- 35.1.62 `CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 781

35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if `sourceCount` is 0 or source array is empty.

`sourceCount` can be -2 to use array count - `sourceIndex`.

`destIndex` lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 `CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 768

- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 768
- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772

35.1. MODULE SORTMBS

771

- 35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 768
- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774

- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778

- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780

35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Example:

```
Var n() As Integer = Array(1,2)
```

```
Var a() As Integer = Array(3,4)
```

```
Call n.CopyMBS(a)
```

```
Break // check n in debugger
```

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 `CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 768
- 35.1.49 `CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 769
- 35.1.50 `CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 770
- 35.1.51 `CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 771
- 35.1.53 `CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 773
- 35.1.54 `CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 774
- 35.1.57 `CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 776
- 35.1.60 `CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 779
- 35.1.61 `CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 780
- 35.1.62 `CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 781

35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 `CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 768
- 35.1.50 `CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 770
- 35.1.51 `CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 771
- 35.1.52 `CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 772
- 35.1.53 `CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 773
- 35.1.54 `CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 774
- 35.1.56 `CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 776
- 35.1.60 `CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 779
- 35.1.61 `CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 780
- 35.1.62 `CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 781

35.1.58 `CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)`

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 `CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 768
- 35.1.50 `CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 770
- 35.1.53 `CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 773
- 35.1.54 `CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 774
- 35.1.55 `CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 774
- 35.1.56 `CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 776
- 35.1.57 `CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 776
- 35.1.59 `CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 778
- 35.1.61 `CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 780
- 35.1.62 `CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 781

35.1.59 `CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)`

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if `sourceCount` is 0 or source array is empty.

`sourceCount` can be -2 to use array count - `sourceIndex`.

`destIndex` lets you pick where to store in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 `CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)` 768

- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770
- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780
- 35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 781

35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 768
- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.50 CopyMBS(extends source() as Currency, dest() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 770

- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780

35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 768
- 35.1.51 CopyMBS(extends source() as Date, dest() as Date, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 771
- 35.1.52 CopyMBS(extends source() as DateTime, dest() as DateTime, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 772
- 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- 35.1.54 CopyMBS(extends source() as Int32, dest() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774

- 35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779

35.1.62 CopyMBS(extends source() as Variant, dest() as Variant, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Copies the content of one array to another array.

Notes: Does nothing if sourceCount is 0 or source array is empty.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to store in the destination array.

Performs bounds checking and may raise an OutOfBoundsException. You may need to resize the destination array to be big enough.

See also:

- 35.1.48 CopyMBS(extends source() as Boolean, dest() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 768
- 35.1.49 CopyMBS(extends source() as Color, dest() as Color, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 769
- 35.1.53 CopyMBS(extends source() as Double, dest() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 773
- 35.1.55 CopyMBS(extends source() as Int64, dest() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 774
- 35.1.56 CopyMBS(extends source() as Object, dest() as Object, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.57 CopyMBS(extends source() as Ptr, dest() as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 776
- 35.1.58 CopyMBS(extends source() as Single, dest() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 777

- 35.1.59 CopyMBS(extends source() as String, dest() as String, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 778
- 35.1.60 CopyMBS(extends source() as UInt32, dest() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 779
- 35.1.61 CopyMBS(extends source() as UInt64, dest() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, destIndex as Integer = 0) 780

35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Example:

```
Var values1() As Integer = Array(546, 453, 345, 453, 878)
Var values2() As Boolean = Array(True, False, True, False, True)
```

```
Var c1 As Integer = values1.CountValueMBS(453) // 2
Var c2 As Integer = values2.CountValueMBS(True) // 3
Var c3 As Integer = values2.CountValueMBS(False) // 2
```

```
MessageBox c1.ToString+EndOfLine+c2.ToString+EndOfLine+c3.ToString
```

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787

- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785

- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Example:

```
Var values1() As Integer = Array(546, 453, 345, 453, 878)
Var values2() As Boolean = Array(True, False, True, False, True)
```

```
Var c1 As Integer = values1.CountValueMBS(453) // 2
Var c2 As Integer = values2.CountValueMBS(True) // 3
Var c3 As Integer = values2.CountValueMBS(False) // 2
```

```
MessageBox c1.ToString+EndOfLine+c2.ToString+EndOfLine+c3.ToString
```

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786

- 35.1.69 `CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 787
- 35.1.70 `CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 787
- 35.1.71 `CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 788
- 35.1.72 `CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 789

35.1.68 `CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer`

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 `CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 782
- 35.1.64 `CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 783
- 35.1.65 `CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 783
- 35.1.66 `CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 784
- 35.1.67 `CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 785
- 35.1.69 `CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 787
- 35.1.70 `CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 787
- 35.1.71 `CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 788
- 35.1.72 `CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer` 789

35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Example:

```
Var s() As String = Array("a", "b", "a", "b", "c")
Var c As Integer = s.CountValueMBS("b")
```

MessageBox c.ToString

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783

35.1. MODULE SORTMBS

789

- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 789

35.1.72 CountValueMBS(extends source() as UInt64, value as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Counts how often the given value is inside the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

See also:

- 35.1.63 CountValueMBS(extends source() as Boolean, value as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 782
- 35.1.64 CountValueMBS(extends source() as Currency, value as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.65 CountValueMBS(extends source() as Double, value as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 783
- 35.1.66 CountValueMBS(extends source() as Int32, value as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 784
- 35.1.67 CountValueMBS(extends source() as Int64, value as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 785
- 35.1.68 CountValueMBS(extends source() as Ptr, value as Ptr, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 786
- 35.1.69 CountValueMBS(extends source() as Single, value as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.70 CountValueMBS(extends source() as String, value as String, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 787
- 35.1.71 CountValueMBS(extends source() as UInt32, value as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Integer 788

35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
 destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.
 See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Example:

```
Const u = 99999
```

```
Var v1(u) As Double
```

```
Var v2(u) As Double
```



```

For i As Integer = 0 To u
v1(i) = i
v2(i) = i
Next

// should be equal
Var b1 As Boolean = EqualsArrayMBS(v1, v2)

v1(123) = 0

// should not be equal
Var b2 As Boolean = EqualsArrayMBS(v1, v2)

// should be equal
Var b3 As Boolean = EqualsArrayMBS(v1, v2, 200, -2, 200)
Var b4 As Boolean = EqualsArrayMBS(v1, v2, 0, 100, 0)

// and this should be false
Var b5 As Boolean = EqualsArrayMBS(v1, v2, 0, 200, 0)
Var b6 As Boolean = EqualsArrayMBS(v1, v2, 1) // <- offset one in first array

Break

```

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.
See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796

- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798

- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Example:

```
Const u = 99999

Var v1(u) As Integer
Var v2(u) As Integer

For i As Integer = 0 To u
    v1(i) = i
    v2(i) = i
Next

// should be equal
Var b1 As Boolean = EqualsArrayMBS(v1, v2)

v1(123) = 0

// should not be equal
Var b2 As Boolean = EqualsArrayMBS(v1, v2)

// should be equal
Var b3 As Boolean = EqualsArrayMBS(v1, v2, 200, -2, 200)
Var b4 As Boolean = EqualsArrayMBS(v1, v2, 0, 100, 0)

// and this should be false
Var b5 As Boolean = EqualsArrayMBS(v1, v2, 0, 200, 0)
Var b6 As Boolean = EqualsArrayMBS(v1, v2, 1) // <- offset one in first array

Break
```

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`.

See also:

- 35.1.73 `EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 790
- 35.1.74 `EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 790
- 35.1.75 `EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 791
- 35.1.76 `EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 792
- 35.1.77 `EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 794
- 35.1.79 `EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 796
- 35.1.80 `EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 797
- 35.1.81 `EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 798
- 35.1.82 `EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 800
- 35.1.83 `EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 800

35.1.79 `EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`.

See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790

- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Example:

```
Const u = 99999
```

```
Var v1(u) As String
```

```
Var v2(u) As String
```

```
For i As Integer = 0 To u
```

```
  v1(i) = i.ToString
```

```
  v2(i) = i.ToString
```

```
Next
```

```
// these will have same pointers, but we do compare string content if not.
```

```
v1(0) = v2(0)
```

```
// should be equal
```

```
Var b1 As Boolean = EqualsArrayMBS(v1, v2)
```

```
v1(123) = "test"
```

```

// should not be equal
Var b2 As Boolean = EqualsArrayMBS(v1, v2)

// should be equal
Var b3 As Boolean = EqualsArrayMBS(v1, v2, 200, -2, 200)
Var b4 As Boolean = EqualsArrayMBS(v1, v2, 0, 100, 0)

// and this should be false
Var b5 As Boolean = EqualsArrayMBS(v1, v2, 0, 200, 0)
Var b6 As Boolean = EqualsArrayMBS(v1, v2, 1) // <- offet one in first array

Break

```

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.
See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.83 EqualsArrayMBS(array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.73 EqualsArrayMBS(array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.74 EqualsArrayMBS(array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 790
- 35.1.75 EqualsArrayMBS(array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 791
- 35.1.76 EqualsArrayMBS(array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 792
- 35.1.77 EqualsArrayMBS(array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 794
- 35.1.78 EqualsArrayMBS(array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 795
- 35.1.79 EqualsArrayMBS(array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 796
- 35.1.80 EqualsArrayMBS(array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 797
- 35.1.81 EqualsArrayMBS(array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 798
- 35.1.82 EqualsArrayMBS(array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 800

35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
 destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.
 See also:

- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
 destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`.

See also:

- 35.1.84 `EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 801
- 35.1.86 `EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 803
- 35.1.87 `EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 804
- 35.1.88 `EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 805
- 35.1.89 `EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 806
- 35.1.90 `EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 807
- 35.1.91 `EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 808
- 35.1.92 `EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 809
- 35.1.93 `EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 810
- 35.1.94 `EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 811

35.1.86 `EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean`

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

`SourceCount` can be -2 to use array count - `sourceIndex`.

`destIndex` lets you pick where to start in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`.

See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802

35.1. MODULE SORTMBS	805
• 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	803
• 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	805
• 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	806
• 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	807
• 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	808
• 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	809
• 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	810
• 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean	811

35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804

- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Example:

```
Const u = 99999

Var v1(u) As Integer
Var v2(u) As Integer

For i As Integer = 0 To u
    v1(i) = i
    v2(i) = i
Next

// should be equal
Var b1 As Boolean = v1.EqualsMBS(v2)

v1(123) = 0

// should not be equal
Var b2 As Boolean = v1.EqualsMBS(v2)

// should be equal
Var b3 As Boolean = v1.EqualsMBS(v2, 200, -2, 200)
Var b4 As Boolean = v1.EqualsMBS(v2, 0, 100, 0)
```

```
// and this should be false
Var b5 As Boolean = v1.EqualsMBS(v2, 0, 200, 0)
Var b6 As Boolean = v1.EqualsMBS(v2, 1) // <- offset one in first array
```

Break

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.
See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.
See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.
destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`.

See also:

- 35.1.84 `EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 801
- 35.1.85 `EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 802
- 35.1.86 `EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 803
- 35.1.87 `EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 804
- 35.1.88 `EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 805
- 35.1.89 `EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 806
- 35.1.90 `EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 807
- 35.1.91 `EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 808
- 35.1.93 `EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 810
- 35.1.94 `EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean` 811

35.1.93 `EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean`

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

`SourceCount` can be -2 to use array count - `sourceIndex`.

`destIndex` lets you pick where to start in the destination array.

Performs bounds checking and may raise an `OutOfBoundsException`.

See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802
- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 811

35.1.94 EqualsMBS(extends array1() as UInt64, array2() as UInt64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Compares the content of one array to another array.

Notes: Returns true if they are equal in the given range.

SourceCount can be -2 to use array count - sourceIndex.

destIndex lets you pick where to start in the destination array.

Performs bounds checking and may raise an OutOfBoundsException.

See also:

- 35.1.84 EqualsMBS(extends array1() as Boolean, array2() as Boolean, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 801
- 35.1.85 EqualsMBS(extends array1() as Color, array2() as Color, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 802

- 35.1.86 EqualsMBS(extends array1() as Currency, array2() as Currency, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 803
- 35.1.87 EqualsMBS(extends array1() as Double, array2() as Double, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 804
- 35.1.88 EqualsMBS(extends array1() as Int32, array2() as Int32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 805
- 35.1.89 EqualsMBS(extends array1() as Int64, array2() as Int64, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 806
- 35.1.90 EqualsMBS(extends array1() as Ptr, array2() as Ptr, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 807
- 35.1.91 EqualsMBS(extends array1() as Single, array2() as Single, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 808
- 35.1.92 EqualsMBS(extends array1() as String, array2() as String, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 809
- 35.1.93 EqualsMBS(extends array1() as UInt32, array2() as UInt32, array1offset as Integer = 0, count as Integer = -2, array2offset as Integer = 0) as Boolean 810

35.1.95 ReverseMBS(extends theArray() as Boolean)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.96 ReverseMBS(extends theArray() as Color)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.97 ReverseMBS(extends theArray() as Currency)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.104 ReverseMBS(extends theArray() as Ptr) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819

35.1.98 ReverseMBS(extends theArray() as Date)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.104 ReverseMBS(extends theArray() as Ptr) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819

35.1.99 ReverseMBS(extends theArray() as DateTime)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.104 ReverseMBS(extends theArray() as Ptr) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.100 ReverseMBS(extends theArray() as Double)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

Example:

```
Var a() As Double = Array(1.1,2.2,3.3,4.4,5.5)
```

```
a.ReverseMBS
```

```
Break // check in debugger
```

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.101 ReverseMBS(extends theArray() as Int32)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815

- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.104 ReverseMBS(extends theArray() as Ptr) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.102 ReverseMBS(extends theArray() as Int64)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

Example:

```
Var a() As Integer = Array(1,2,3,4,5)
```

```
a.ReverseMBS
```

```
Break // check in debugger
```

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.103 ReverseMBS(extends theArray() as Object)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.104 ReverseMBS(extends theArray() as Ptr)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819

35.1.105 ReverseMBS(extends theArray() as Single)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.95 ReverseMBS(extends theArray() as Boolean) 812
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.102 ReverseMBS(extends theArray() as Int64) 816
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.106 ReverseMBS(extends theArray() as String)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

Example:

```
Var a() As String = Array("a", "b", "c", "d")
```

```
a.ReverseMBS
```

```
Break // check in debugger
```

See also:

- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.99 ReverseMBS(extends theArray() as DateTime) 814
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815

35.1. <i>MODULE SORTMBS</i>	819
• 35.1.102 ReverseMBS(extends theArray() as Int64)	816
• 35.1.104 ReverseMBS(extends theArray() as Ptr)	817
• 35.1.105 ReverseMBS(extends theArray() as Single)	818
• 35.1.107 ReverseMBS(extends theArray() as UInt32)	819
• 35.1.108 ReverseMBS(extends theArray() as UInt64)	819

35.1.107 ReverseMBS(extends theArray() as UInt32)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

• 35.1.95 ReverseMBS(extends theArray() as Boolean)	812
• 35.1.97 ReverseMBS(extends theArray() as Currency)	813
• 35.1.98 ReverseMBS(extends theArray() as Date)	814
• 35.1.99 ReverseMBS(extends theArray() as DateTime)	814
• 35.1.101 ReverseMBS(extends theArray() as Int32)	815
• 35.1.102 ReverseMBS(extends theArray() as Int64)	816
• 35.1.104 ReverseMBS(extends theArray() as Ptr)	817
• 35.1.106 ReverseMBS(extends theArray() as String)	818
• 35.1.108 ReverseMBS(extends theArray() as UInt64)	819
• 35.1.109 ReverseMBS(extends theArray() as Variant)	820

35.1.108 ReverseMBS(extends theArray() as UInt64)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

• 35.1.98 ReverseMBS(extends theArray() as Date)	814
• 35.1.99 ReverseMBS(extends theArray() as DateTime)	814
• 35.1.100 ReverseMBS(extends theArray() as Double)	815
• 35.1.101 ReverseMBS(extends theArray() as Int32)	815

- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.104 ReverseMBS(extends theArray() as Ptr) 817
- 35.1.105 ReverseMBS(extends theArray() as Single) 818
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.109 ReverseMBS(extends theArray() as Variant) 820

35.1.109 ReverseMBS(extends theArray() as Variant)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reverses the order of items in array.

See also:

- 35.1.96 ReverseMBS(extends theArray() as Color) 813
- 35.1.97 ReverseMBS(extends theArray() as Currency) 813
- 35.1.98 ReverseMBS(extends theArray() as Date) 814
- 35.1.100 ReverseMBS(extends theArray() as Double) 815
- 35.1.101 ReverseMBS(extends theArray() as Int32) 815
- 35.1.103 ReverseMBS(extends theArray() as Object) 817
- 35.1.104 ReverseMBS(extends theArray() as Ptr) 817
- 35.1.106 ReverseMBS(extends theArray() as String) 818
- 35.1.107 ReverseMBS(extends theArray() as UInt32) 819
- 35.1.108 ReverseMBS(extends theArray() as UInt64) 819

35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a boolean array.

Example:

```
Var test() As Boolean = Array(False, True, False, True, False)
```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
SortArrayMBS test, True
```

```
Break // see in debugger
```

Notes: Does nothing for arrays with theArray.count <2.
Pass true for descending to reverse the order.
See also:

- 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false) 822
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 831
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 837
- 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 839
- 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 841
- 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 841

35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a boolean array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.
Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false) 822

- 35.1.115 `SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)` 825
- 35.1.116 `SortArrayMBS(theArray() as Date, descending as boolean = false)` 825
- 35.1.118 `SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)` 828
- 35.1.120 `SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)` 829
- 35.1.121 `SortArrayMBS(theArray() as Int32, descending as boolean = false)` 830
- 35.1.122 `SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)` 831
- 35.1.129 `SortArrayMBS(theArray() as String, descending as boolean = false)` 836
- 35.1.130 `SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)` 837
- 35.1.131 `SortArrayMBS(theArray() as UInt32, descending as boolean = false)` 838

35.1.112 `SortArrayMBS(theArray() as Color, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a color array.

Example:

```
Var test() As Color = Array(&cFF000, &c00FF00, &c0000FF, &c000000, &cFFFFFF)
```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
```

```
SortArrayMBS test, True
```

```
Break // see in debugger
```

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

See also:

- 35.1.114 `SortArrayMBS(theArray() as Currency, descending as boolean = false)` 824
- 35.1.115 `SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)` 825

35.1. MODULE SORTMBS	823
• 35.1.120 SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	829
• 35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)	831
• 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false)	831
• 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false)	836
• 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)	837
• 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)	839
• 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)	841
• 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)	841

35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a color array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

• 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)	821
• 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false)	822
• 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false)	826
• 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false)	830
• 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false)	831
• 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false)	833
• 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)	834

- 35.1.127 `SortArrayMBS(theArray() as Single, descending as boolean = false)` 835
- 35.1.133 `SortArrayMBS(theArray() as UInt64, descending as boolean = false)` 840
- 35.1.134 `SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)` 841

35.1.114 `SortArrayMBS(theArray() as Currency, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a currency array.

Example:

```
Var test() As Currency
```

```
test.Append 1.2
```

```
test.Append -1.2
```

```
test.Append 12345
```

```
test.Append 3456
```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
```

```
SortArrayMBS test, True
```

```
Break // see in debugger
```

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

See also:

- 35.1.111 `SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)` 821
- 35.1.113 `SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)` 823
- 35.1.115 `SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)` 825
- 35.1.117 `SortArrayMBS(theArray() as DateTime, descending as boolean = false)` 826
- 35.1.120 `SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)` 829
- 35.1.121 `SortArrayMBS(theArray() as Int32, descending as boolean = false)` 830
- 35.1.123 `SortArrayMBS(theArray() as Int64, descending as boolean = false)` 831

35.1. MODULE SORTMBS	825
• 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)	832
• 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false)	838
• 35.1.133 SortArrayMBS(theArray() as UInt64, descending as boolean = false)	840

35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a currency array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

• 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false)	822
• 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false)	825
• 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false)	826
• 35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false)	828
• 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false)	831
• 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)	832
• 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false)	835
• 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false)	836
• 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false)	838
• 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)	841

35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Sorts a date array.

Example:

```

Var test() As Date
test.Append New date
test.Append New date(2023, 4, 5)
test.Append New date(2011, 3, 10)
test.Append New date(2033, 6, 22)

```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
SortArrayMBS test, True
```

```
Break // see in debugger
```

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 821
- 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 825
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830
- 35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 831
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835
- 35.1.133 SortArrayMBS(theArray() as UInt64, descending as boolean = false) 840

35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a DateTime array.

Example:

```

Var test() As DateTime
test.Append DateTime.Now
test.Append DateTime.FromString("2011-04-24")
test.Append DateTime.FromString("2022-03-21")
test.Append DateTime.FromString("2033-10-10")

```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
SortArrayMBS test, True
```

```
Break // see in debugger
```

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 821
- 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 825
- 35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 828
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 837
- 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false) 838
- 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 839
- 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 841

35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a datetime array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false) 825
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false) 828
- 35.1.120 SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false) 829
- 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835
- 35.1.128 SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 836
- 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 837
- 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false) 838

35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a double array.

Example:

```
Var test() As Double = Array(5.0, -2, 3, -4, 1)
```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
```

SortArrayMBS test, [True](#)

[Break](#) // see in debugger

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 831
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835
- 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false) 838

35.1.120 SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a double array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false) 825
- 35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 828

- 35.1.122 `SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)` 831
- 35.1.124 `SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)` 832
- 35.1.126 `SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)` 834
- 35.1.127 `SortArrayMBS(theArray() as Single, descending as boolean = false)` 835
- 35.1.129 `SortArrayMBS(theArray() as String, descending as boolean = false)` 836
- 35.1.133 `SortArrayMBS(theArray() as UInt64, descending as boolean = false)` 840
- 35.1.134 `SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)` 841

35.1.121 `SortArrayMBS(theArray() as Int32, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an Int32 array.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

See also:

- 35.1.110 `SortArrayMBS(theArray() as Boolean, descending as boolean = false)` 820
- 35.1.111 `SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)` 821
- 35.1.113 `SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)` 823
- 35.1.116 `SortArrayMBS(theArray() as Date, descending as boolean = false)` 825
- 35.1.122 `SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)` 831
- 35.1.125 `SortArrayMBS(theArray() as Ptr, descending as boolean = false)` 833
- 35.1.128 `SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)` 836
- 35.1.131 `SortArrayMBS(theArray() as UInt32, descending as boolean = false)` 838
- 35.1.133 `SortArrayMBS(theArray() as UInt64, descending as boolean = false)` 840
- 35.1.135 `SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)` 841

35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an Int32 array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 825
- 35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false) 828
- 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830
- 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false) 831
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 841
- 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 841

35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an Int64 array.

Example:

```
Var test() As Integer = Array(5, -2, 3, -4, 1)
```

```
SortArrayMBS test
```

```
Break // see in debugger
```

```
// now reverse it
SortArrayMBS test, True

Break // see in debugger
```

Notes: Does nothing for arrays with theArray.count <2.
Pass true for descending to reverse the order.
See also:

- 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false) 822
- 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 825
- 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false) 825
- 35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 828
- 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835
- 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 837

35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an Int64 array using a delegate to compare.

Example:

```
Public Function CompareNumber(v1 as Integer, v2 as Integer) As integer
// we compare the absolute values, so 3 and -3 are next to each other in the result
v1 = Abs(v1)
v2 = Abs(v2)

If v1 = v2 Then
Return 0
ElseIf v1 <v2 Then
```



```

Return -1
Else
Return 1
End If
End Function

```

```

Sub Test
Var testInteger() As Integer = Array(1,3,6,1,2,-3,0,9)
SortArrayMBS testInteger, AddressOf CompareNumber

```

```

Break // see in debugger
End Sub

```

Notes: Does nothing for arrays with theArray.count < 2.
Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 821
- 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false) 825
- 35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 828
- 35.1.122 SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 831
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 837
- 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false) 838
- 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 839

35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a ptr array.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

See also:

- 35.1.112 `SortArrayMBS(theArray() as Color, descending as boolean = false)` 822
- 35.1.113 `SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)` 823
- 35.1.115 `SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)` 825
- 35.1.116 `SortArrayMBS(theArray() as Date, descending as boolean = false)` 825
- 35.1.120 `SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)` 829
- 35.1.122 `SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)` 831
- 35.1.128 `SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)` 836
- 35.1.130 `SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)` 837
- 35.1.133 `SortArrayMBS(theArray() as UInt64, descending as boolean = false)` 840
- 35.1.134 `SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)` 841

35.1.126 `SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a Ptr array using a delegate to compare.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.111 `SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)` 821
- 35.1.113 `SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)` 823

35.1. MODULE SORTMBS	835
• 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	825
• 35.1.120 SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	829
• 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false)	830
• 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false)	831
• 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false)	835
• 35.1.128 SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	836
• 35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)	837
• 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)	841

35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a single array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

• 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false)	822
• 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false)	826
• 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false)	830
• 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false)	833
• 35.1.128 SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	836
• 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false)	836
• 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false)	838
• 35.1.133 SortArrayMBS(theArray() as UInt64, descending as boolean = false)	840
• 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)	841
• 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)	841

35.1.128 SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a single array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 821
- 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- 35.1.116 SortArrayMBS(theArray() as Date, descending as boolean = false) 825
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- 35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false) 838
- 35.1.133 SortArrayMBS(theArray() as UInt64, descending as boolean = false) 840
- 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 841
- 35.1.135 SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 841

35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a string array.

Example:

```
Var testString() As String = Array("c", "b", "e", "d", "a")
```

```
SortArrayMBS testString
```

Break // see in debugger

```
// now reverse it
SortArrayMBS testString, True
```

Break // see in debugger

Notes: Does nothing for arrays with theArray.count <2.
Pass true for descending to reverse the order.

See also NSStringArraySortMBS function.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false) 822
- 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.118 SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 828
- 35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false) 828
- 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835

35.1.130 SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a string array using a delegate to compare.

Example:

```
Public Function CompareString2ndWord(s1 as string, s2 as string) As integer
// sort by second word
```

```
s1 = s1.NthField(" ",2)
s2 = s2.NthField(" ",2)
```

```
Return s1.Compare(s2)
End Function
```

```
Sub Test
Var testString() As String = Array("Hello World", "Hallo Leute", "World Test", "First Entry", "abc def",
"abc xyz")
SortArrayMBS testString, AddressOf CompareString2ndWord

Break // see in debugger
End Sub
```

Notes: Does nothing for arrays with theArray.count <2.
Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.111 SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 821
- 35.1.112 SortArrayMBS(theArray() as Color, descending as boolean = false) 822
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.119 SortArrayMBS(theArray() as Double, descending as boolean = false) 828
- 35.1.123 SortArrayMBS(theArray() as Int64, descending as boolean = false) 831
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.126 SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 834
- 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 839

35.1.131 SortArrayMBS(theArray() as UInt32, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an UInt32 array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.110 SortArrayMBS(theArray() as Boolean, descending as boolean = false) 820
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.120 SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false) 829
- 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830
- 35.1.124 SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 832
- 35.1.125 SortArrayMBS(theArray() as Ptr, descending as boolean = false) 833
- 35.1.127 SortArrayMBS(theArray() as Single, descending as boolean = false) 835
- 35.1.129 SortArrayMBS(theArray() as String, descending as boolean = false) 836
- 35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 839
- 35.1.134 SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 841

35.1.132 SortArrayMBS(theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an UInt32 array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count < 2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.113 SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 823
- 35.1.114 SortArrayMBS(theArray() as Currency, descending as boolean = false) 824
- 35.1.115 SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 825
- 35.1.117 SortArrayMBS(theArray() as DateTime, descending as boolean = false) 826
- 35.1.121 SortArrayMBS(theArray() as Int32, descending as boolean = false) 830

- 35.1.126 `SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)` 834
- 35.1.129 `SortArrayMBS(theArray() as String, descending as boolean = false)` 836
- 35.1.130 `SortArrayMBS(theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)` 837
- 35.1.131 `SortArrayMBS(theArray() as UInt32, descending as boolean = false)` 838
- 35.1.134 `SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)` 841

35.1.133 `SortArrayMBS(theArray() as UInt64, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an UInt64 array.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

See also:

- 35.1.111 `SortArrayMBS(theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)` 821
- 35.1.112 `SortArrayMBS(theArray() as Color, descending as boolean = false)` 822
- 35.1.117 `SortArrayMBS(theArray() as DateTime, descending as boolean = false)` 826
- 35.1.118 `SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)` 828
- 35.1.120 `SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)` 829
- 35.1.121 `SortArrayMBS(theArray() as Int32, descending as boolean = false)` 830
- 35.1.123 `SortArrayMBS(theArray() as Int64, descending as boolean = false)` 831
- 35.1.126 `SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)` 834
- 35.1.129 `SortArrayMBS(theArray() as String, descending as boolean = false)` 836
- 35.1.131 `SortArrayMBS(theArray() as UInt32, descending as boolean = false)` 838

35.1.134 `SortArrayMBS(theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an UInt64 array using a delegate to compare.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.110 `SortArrayMBS(theArray() as Boolean, descending as boolean = false)` 820
- 35.1.112 `SortArrayMBS(theArray() as Color, descending as boolean = false)` 822
- 35.1.115 `SortArrayMBS(theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)` 825
- 35.1.118 `SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)` 828
- 35.1.119 `SortArrayMBS(theArray() as Double, descending as boolean = false)` 828
- 35.1.122 `SortArrayMBS(theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)` 831
- 35.1.124 `SortArrayMBS(theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)` 832
- 35.1.128 `SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)` 836
- 35.1.129 `SortArrayMBS(theArray() as String, descending as boolean = false)` 836
- 35.1.135 `SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)` 841

35.1.135 `SortArrayMBS(theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)`

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts a variant array using a delegate to compare.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.113 `SortArrayMBS(theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)` 823
- 35.1.114 `SortArrayMBS(theArray() as Currency, descending as boolean = false)` 824
- 35.1.116 `SortArrayMBS(theArray() as Date, descending as boolean = false)` 825
- 35.1.118 `SortArrayMBS(theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)` 828
- 35.1.120 `SortArrayMBS(theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)` 829
- 35.1.123 `SortArrayMBS(theArray() as Int64, descending as boolean = false)` 831
- 35.1.125 `SortArrayMBS(theArray() as Ptr, descending as boolean = false)` 833
- 35.1.126 `SortArrayMBS(theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)` 834
- 35.1.128 `SortArrayMBS(theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)` 836
- 35.1.129 `SortArrayMBS(theArray() as String, descending as boolean = false)` 836

35.1.136 `SortMBS(extends theArray() as Boolean, descending as boolean = false)`

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with `theArray.count < 2`.

Pass true for descending to reverse the order.

See also:

- 35.1.137 `SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)` 843
- 35.1.143 `SortMBS(extends theArray() as DateTime, descending as boolean = false)` 847
- 35.1.144 `SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)` 848
- 35.1.147 `SortMBS(extends theArray() as Int32, descending as boolean = false)` 851
- 35.1.150 `SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)` 853

- 35.1. *MODULE SORTMBS* 843
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859

35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860

35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 846
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.145 SortMBS(extends theArray() as Double, descending as boolean = false) 848
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false) 854
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859
- 35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 860

35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843

35.1. MODULE SORTMBS	845
• 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	846
• 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false)	846
• 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)	848
• 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	850
• 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)	851
• 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)	852
• 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	856
• 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false)	858
• 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false)	860

35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

• 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false)	842
• 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	846
• 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false)	846
• 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)	848
• 35.1.145 SortMBS(extends theArray() as Double, descending as boolean = false)	848
• 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)	851
• 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)	852
• 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	856
• 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)	857
• 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false)	860

35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false) 850
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 853
- 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false) 855
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 860

35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842

35.1. MODULE SORTMBS	847
• 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	846
• 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false)	847
• 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	850
• 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)	851
• 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)	851
• 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)	853
• 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false)	854
• 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)	859
• 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false)	860

35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 853
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854

- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860
- 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 861

35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851
- 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false) 855
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860

35.1.145 SortMBS(extends theArray() as Double, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Example:


```
// disable background threads, which would slow us down
#pragma BackgroundTasks False

Var a(1000000) As Double
Var b(1000000) As Double
Var c(1000000) As Double

Var u As Integer = a.ubound
For i As Integer = 0 To u
  Var n As Integer = i Mod 1000
  a(i) = n
  b(i) = n
  c(i) = n
Next

Var m1 As Double = Microseconds
a.Sort
Var m2 As Double = Microseconds
b.SortMBS
Var m3 As Double = Microseconds
c.SortMBS True // reverse
Var m4 As Double = Microseconds

Var XojoSort As Double = (m2 - m1) / 1000.0 // e.g. 142
Var MBSSort As Double = (m3 - m2) / 1000.0 // e.g. 128
Var MBSSortReverse As Double = (m4 - m3) / 1000.0 // e.g. 120

MessageBox "Xojo: " + XojoSort.ToString + EndOfLine + "MBS Sort: " + MBSSort.ToString + EndOfLine + "MBS
Sort reverse: " + MBSSortReverse.ToString
Break
```

Notes: Does nothing for arrays with theArray.count < 2.
 Pass true for descending to reverse the order.
 See also:

- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 846
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851

- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 853
- 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false) 855
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859

35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842
- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860

35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842
- 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845
- 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false) 850
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false) 854
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858

35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false) 853
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860
- 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 861

35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Example:

```
Var test() As Integer = Array(5, -2, 3, -4, 1)
```

```
test.SortMBS
```

```
Break // see in debugger
```

```
// now reverse it
test.SortMBS True
```

```
Break // see in debugger
```

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 846

35.1. MODULE SORTMBS	853
• 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)	848
• 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	850
• 35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)	853
• 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false)	854
• 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)	854
• 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)	859
• 35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)	860

35.1.150 SortMBS(extends theArray() as Int64, theDelegate as SortVariantDelegateInt64MBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

• 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)	843
• 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false)	844
• 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)	852
• 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)	854
• 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	856
• 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false)	857
• 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)	857

- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860
- 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 861

35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.136 SortMBS(extends theArray() as Boolean, descending as boolean = false) 842
- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false) 850
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false) 855
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860

35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851
- 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false) 854
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859

35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false) 854
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856

- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858
- 35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false) 860

35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845
- 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 846
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false) 854
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859
- 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 861

35.1.155 SortMBS(extends theArray() as String, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also NSStringArraySortMBS function.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 860
- 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false) 861

35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845
- 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false) 846
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false) 854
- 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false) 855
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858
- 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false) 859

35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false) 844
- 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false) 845
- 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false) 846
- 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false) 848
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851

35.1. MODULE SORTMBS	859
• 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)	852
• 35.1.153 SortMBS(extends theArray() as Single, descending as boolean = false)	855
• 35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)	859

35.1.158 SortMBS(extends theArray() as UInt32, theDelegate as SortVariantDelegateUInt32MBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count < 2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

• 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false)	844
• 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)	844
• 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	846
• 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false)	846
• 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false)	847
• 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	850
• 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)	851
• 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)	854
• 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false)	856
• 35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)	861

35.1.159 SortMBS(extends theArray() as UInt64, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843
- 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false) 844
- 35.1.143 SortMBS(extends theArray() as DateTime, descending as boolean = false) 847
- 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false) 851
- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.151 SortMBS(extends theArray() as Ptr, descending as boolean = false) 854
- 35.1.154 SortMBS(extends theArray() as Single, theDelegate as SortVariantDelegateSingleMBS, descending as boolean = false) 856
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false) 857
- 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false) 858

35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

- 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false) 843

35.1. MODULE SORTMBS	861
• 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)	844
• 35.1.140 SortMBS(extends theArray() as Currency, descending as boolean = false)	845
• 35.1.146 SortMBS(extends theArray() as Double, theDelegate as SortVariantDelegateDoubleMBS, descending as boolean = false)	850
• 35.1.147 SortMBS(extends theArray() as Int32, descending as boolean = false)	851
• 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false)	851
• 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false)	852
• 35.1.152 SortMBS(extends theArray() as Ptr, theDelegate as SortVariantDelegatePtrMBS, descending as boolean = false)	854
• 35.1.156 SortMBS(extends theArray() as String, theDelegate as SortVariantDelegateStringMBS, descending as boolean = false)	857
• 35.1.157 SortMBS(extends theArray() as UInt32, descending as boolean = false)	858

35.1.161 SortMBS(extends theArray() as Variant, theDelegate as SortVariantDelegateVariantMBS, descending as boolean = false)

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sorts an array using a delegate to compare.

Notes: Does nothing for arrays with theArray.count <2.

Pass true for descending to reverse the order.

Your delegate should be fast and return 0 if both values are equal, 1 if first value is bigger or -1 if first value is smaller.

See also:

• 35.1.137 SortMBS(extends theArray() as Boolean, theDelegate as SortVariantDelegateBooleanMBS, descending as boolean = false)	843
• 35.1.138 SortMBS(extends theArray() as Color, descending as boolean = false)	844
• 35.1.139 SortMBS(extends theArray() as Color, theDelegate as SortVariantDelegateColorMBS, descending as boolean = false)	844
• 35.1.141 SortMBS(extends theArray() as Currency, theDelegate as SortVariantDelegateCurrencyMBS, descending as boolean = false)	846
• 35.1.142 SortMBS(extends theArray() as Date, descending as boolean = false)	846
• 35.1.144 SortMBS(extends theArray() as DateTime, theDelegate as SortVariantDelegateDateTimeMBS, descending as boolean = false)	848

- 35.1.148 SortMBS(extends theArray() as Int32, theDelegate as SortVariantDelegateInt32MBS, descending as boolean = false) 851
- 35.1.149 SortMBS(extends theArray() as Int64, descending as boolean = false) 852
- 35.1.155 SortMBS(extends theArray() as String, descending as boolean = false) 857
- 35.1.160 SortMBS(extends theArray() as UInt64, theDelegate as SortVariantDelegateUInt64MBS, descending as boolean = false) 860

35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Returns the sum of all values.

See also:

- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 862
- 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 864
- 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 865
- 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 866
- 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 867
- 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868
- 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868

35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Example:

```
Var v() As Currency
```

```
// just sum up some currency values
```

```
v.Add 123.4567
```

```
v.Add 1
```

```
v.add 0
```

```
v.add -56
```

```
v.add 4567
```

```
Var m1 As Double = Microseconds
```

```
Var sum1 As Currency = SumArrayMBS(v, True)
```

```
Var m2 As Double = Microseconds
```

```
Var sum2 As Currency = SumArrayMBS(v, False)
```

```
Var m3 As Double = Microseconds
```

```
Var d1 As Double = m2-m1
```

```
Var d2 As Double = m3-m2
```

```
// overflow checking can make it 100 times slower
```

```
// try to cause an overflow with huge numbers
```

```
v.add 10000000000000.0000
```

```
v.add 20000000000000.0000
```

```
v.add 30000000000000.0000
```

```
v.add 40000000000000.0000
```

```
Var sum3 As Currency = SumArrayMBS(v, False)
```

```
Var sum4 As Currency = SumArrayMBS(v, True)
```

```
Break
```

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 862
- 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 864
- 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 865
- 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2,

- CheckOverflow as Boolean = false) as Int64
866
- 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double
867
- 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64
868
- 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64
868

35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Example:

```

Const u = 99999
Var v(u) As Double

For i As Integer = 0 To u
  v(i) = i
Next

Var m1 As Double = Microseconds
Var sumPlugin As Int64 = SumArrayMBS(v)
Var m2 As Double = Microseconds
Var sumXojo As Double
For i As Integer = 0 To u
  sumXojo = sumXojo + v(i)
next
Var m3 As Double = Microseconds

Var timePlugin As Double = m2-m1
Var timeXojo As Double = m3-m2
// plugin is faster than Xojo code

Break

```

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64
862

- 35.1. *MODULE SORTMBS* 865
- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 862
 - 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 865
 - 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 866
 - 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 867
 - 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868
 - 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868

35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

We use Int64 as result may be bigger than Int32.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 862
- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 862
- 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 864
- 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 866
- 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 867
- 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868
- 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868

35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Example:

```

Var v() As Int64

// just sum up some currency values
v.Add 1234567
v.Add 1
v.add 0
v.add -56
v.add 4567

Var m1 As Double = Microseconds
Var sum1 As Int64 = SumArrayMBS(v, True)
Var m2 As Double = Microseconds
Var sum2 As Int64 = SumArrayMBS(v, False)
Var m3 As Double = Microseconds

Var d1 As Double = m2-m1
Var d2 As Double = m3-m2
// overflow checking can make it 100 times slower

// try to cause an overflow with huge numbers
v.add 1000000000000000000
v.add 2000000000000000000
v.add 3000000000000000000
v.add 4000000000000000000

Var sum3 As Int64 = SumArrayMBS(v, False)
Var sum4 As Int64 = SumArrayMBS(v, True)

Break

```

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 862
- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer =

35.1. MODULE SORTMBS	867
-2, CheckOverflow as Boolean = false) as Currency	862
• 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double	864
• 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64	865
• 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double	867
• 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64	868
• 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64	868

35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 862
- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 862
- 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 864
- 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 865
- 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 866
- 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868
- 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868

35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 862
- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 862
- 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 864
- 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 865
- 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 866
- 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 867
- 35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868

35.1.169 SumArrayMBS(source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Example:

```
Const u = 99999
Var v(u) As UInt64
```

```
For i As Integer = 0 To u
v(i) = i
Next
```

```
Var m1 As Double = Microseconds
Var sumPlugin As UInt64 = SumArrayMBS(v, False)
```

```

Var m2 As Double = Microseconds
Var sumPluginOverflowChecked As UInt64 = SumArrayMBS(v, True)
Var m3 As Double = Microseconds
Var sumXojo As UInt64
For i As Integer = 0 To u
sumXojo = sumXojo + v(i)
Next
Var m4 As Double = Microseconds

```

```

Var timePlugin As Double = m2-m1
Var timePluginOverflowChecked As Double = m3-m2
Var timeXojo As Double = m4-m3
// plugin is faster than Xojo code

```

```
Break
```

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.162 SumArrayMBS(source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 862
- 35.1.163 SumArrayMBS(source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 862
- 35.1.164 SumArrayMBS(source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 864
- 35.1.165 SumArrayMBS(source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 865
- 35.1.166 SumArrayMBS(source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 866
- 35.1.167 SumArrayMBS(source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 867
- 35.1.168 SumArrayMBS(source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 868

35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Returns the sum of all values.

See also:

- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
- 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
- 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
- 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
- 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871

- 35.1. *MODULE SORTMBS* 871
- 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
 - 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
 - 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
 - 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
- 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
- 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
- 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
- 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871
- 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
- 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
- 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
- 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871

- 35.1. *MODULE SORTMBS* 873
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
 - 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
 - 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
 - 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
- 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
- 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873
- 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
- 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
- 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
- 35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 874

35.1.177 SumMBS(extends source() as UInt64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64

Plugin Version: 24.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sums up all the numbers in the array.

Notes: You can limit range by passing in a source index and source count.

If sourceCount is -2, we sum up whole array - sourceIndex.

Pass true for overflow check to detect overflows for each addition. Otherwise pass false for better performance.

Returns the sum of all values.

See also:

- 35.1.170 SumMBS(extends source() as Boolean, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 869
- 35.1.171 SumMBS(extends source() as Currency, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Currency 870
- 35.1.172 SumMBS(extends source() as Double, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 871

- 35.1. *MODULE SORTMBS* 875
- 35.1.173 SumMBS(extends source() as Int32, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Int64 871
 - 35.1.174 SumMBS(extends source() as Int64, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as Int64 872
 - 35.1.175 SumMBS(extends source() as Single, sourceIndex as Integer = 0, sourceCount as Integer = -2) as Double 873
 - 35.1.176 SumMBS(extends source() as UInt32, sourceIndex as Integer = 0, sourceCount as Integer = -2, CheckOverflow as Boolean = false) as UInt64 873

35.1.178 Delegates

35.1.179 SortVariantDelegateBooleanMBS(v1 as Boolean, v2 as Boolean) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two boolean values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

Blog Entries

- [The SortMBS module](#)

35.1.180 SortVariantDelegateColorMBS(v1 as Color, v2 as Color) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two color values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.181 SortVariantDelegateCurrencyMBS(v1 as Currency, v2 as Currency) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two currency values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.182 SortVariantDelegateDateTimeMBS(v1 as DateTime, v2 as DateTime) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two DateTime values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.
Requires Xojo 2019r2 or newer.

35.1.183 SortVariantDelegateDoubleMBS(v1 as Double, v2 as Double) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two double values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.184 SortVariantDelegateInt32MBS(v1 as Int32, v2 as Int32) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two int32 values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.185 SortVariantDelegateInt64MBS(v1 as Int64, v2 as Int64) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two Int64 values.

Example:

```
Public Function CompareNumber(v1 as Integer, v2 as Integer) As integer
// we compare the absolute values, so 3 and -3 are next to each other in the result
v1 = Abs(v1)
v2 = Abs(v2)
```

```
If v1 = v2 Then
Return 0
ElseIf v1 <v2 Then
Return -1
Else
Return 1
End If
End Function
```

```
Sub Test
Var testInteger() As Integer = Array(1,3,6,1,2,-3,0,9)
SortArrayMBS testInteger, AddressOf CompareNumber
```

```
Break // see in debugger
End Sub
```

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.186 SortVariantDelegatePtrMBS(v1 as Ptr, v2 as Ptr) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two Ptr values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.187 SortVariantDelegateSingleMBS(v1 as Single, v2 as Single) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two Single values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.188 SortVariantDelegateStringMBS(v1 as String, v2 as String) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two string values.

Example:

```
Public Function CompareString2ndWord(s1 as string, s2 as string) As integer
// sort by second word
```

```
s1 = s1.NthField(" ",2)
s2 = s2.NthField(" ",2)
```

```
Return s1.Compare(s2)
End Function
```

```
Sub Test
```

```
Var testString() As String = Array("Hello World", "Hallo Leute", "World Test", "First Entry", "abc def",
"abc xyz")
```

```
SortArrayMBS testString, AddressOf CompareString2ndWord
```

```
Break // see in debugger
```

End Sub

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.189 SortVariantDelegateUInt32MBS(v1 as UInt32, v2 as UInt32) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two UInt32 values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.190 SortVariantDelegateUInt64MBS(v1 as UInt64, v2 as UInt64) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two UInt64 values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

35.1.191 SortVariantDelegateVariantMBS(v1 as Variant, v2 as Variant) as Integer

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delegate to compare two variant values.

Notes: Return 0 if equal, 1 if first value is bigger or -1 if first value is smaller.

Chapter 36

Spamsum

36.1 class SpamSumMBS

36.1.1 class SpamSumMBS

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for the spamsum tool.

Notes: Overview

spamsum is a tool for generating and testing signatures on files. The signature is designed to be particularly suitable for producing a result that can be used to compare two emails and see if they are 'similar'. This can provide the core of a SPAM detection system.

The algorithms in spamsum are in two parts. The first part generates a signature which is encoded as a string of ascii characters less than 72 characters long. The second part takes a new signature and a database of existing signatures (actually just a text file with one signature per line) and finds the existing signature that best matches the new signature. A match result in the range of 0 to 100 is generated, where 100 is a perfect match and 0 is a complete mismatch.

Signature Algorithm

The signature algorithm in spamsum has a number of interesting properties that make it especially suitable for SPAM detection.

- non-propagation

In most hash algorithms a change in any part of a plaintext will either change the resulting hash completely or will change all parts of the hash after the part corresponding with the changed plaintext. In the spamsum algorithm only the part of the spamsum signature that corresponds linearly with the changed part of the plaintext will be changed. This means that small changes in any part of the plaintext will leave most of the signature the same. This is essential for SPAM detection as it is common for variants of the same SPAM to have small changes in their body and we need to ensure that the matching algorithm can cope with these changes.

- alignment robustness

Most hash algorithms are very alignment sensitive. If you shift the plaintext by a byte (say by inserting a character at the start) then a completely different hash is generated. The spamsum algorithm is robust to alignment changes, and will automatically re-align the resulting signature after insertions or deletions. This works in combination with the non-propagation property to make spamsum suitable for telling if two emails are 'similar'.

The core of the spamsum algorithm is a rolling hash similar to the rolling hash used in 'rsync'. The rolling hash is used to produce a series of 'reset points' in the plaintext that depend only on the immediate context (with a default context width of seven characters) and not on the earlier or later parts of the plaintext. A stronger hash based on the FNV algorithm is then used to produce hash values of the areas between two reset points. The resulting signature comes from the concatenation of a single character from the FNV hash per reset point.

The frequency of the reset points determines how many characters in the plaintext will be used for each character of output in the signature. At startup spamsum scans the plaintext to determine how many valid input characters are in the plaintext (whitespace is ignored). The algorithm then estimates the reset frequency needed to produce a signature of length 64 and starts producing the signature. If after the signature is produced the result is less than a third of the desired length then the reset frequency is adjusted and the signature re-generated.

Similarity Testing

Once a set of signatures has been generated you need to be able to take a new plaintext and see if it matches one of the signatures. The way this is done is to generate a spamsum signature of the new plaintext then compute a distance measure between each of the existing signatures and the new signature.

The distance measure that spamsum uses is based on the 'string edit distance'. The string edit distance is a measure of how many edit operations are required to take one of the signatures and turn it into the other. In spamsum the 'insert' and 'delete' edit operations are given a weight of 1 while substitution is given a weight of 3 and transposition is given a weight of 5.

The resulting string edit distance is then scaled to produce a 'score' in the range 0-100. A score of 100

indicates a perfect match and a score of 0 indicates a complete mismatch. If the two signatures used a different 'reset frequency' (also known as `block_size`) then the score is automatically set as 0.

The score is weighted so that a value of 50 is a reasonable threshold to use for a 'good match'.

Dual hashes

A significant problem with the above algorithm is the sensitivity to the chosen hash strength of the rolling hash. The initial implementation used a single hash strength chosen based on the file size and rounded to a power of 2. This works, but it means that if the two files being compared cross over a boundary then they will not be able to be compared. To reduce this problem the current implementation chooses two different hash strengths and generates two hashes for each file. This means that the two files will have to have very different lengths for their respected spamsum signatures not to share a common hash strength.

Infrastructure

spamsum is useless without a good quality database of signatures for known spam. I am hoping that the spamsum algorithm will be incorporated into an online system for capturing known SPAM (such as razor).

Author

spamsum was written by Andrew Tridgell `tridge @ samba.org`

36.1.2 Methods

36.1.3 Match(sum1 as string, sum2 as string) as Integer

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a percent value representing

Example:

```
Var s as SpamSumMBS
```

```
Var t1,t2 as string
```

```
s=new SpamSumMBS
```

```
t1=S.Spamsum("Hello Welt, wie geht es? Mir geht es manchmal gut.",0,0)
```

```
t2=S.Spamsum("Hallo Welt, wie geht es? Mir geht es manchmal gut.",0,0)
```

```
MsgBox t1+" "+t2+" "+str(s.Match(t1,t2))  
// 17 percent here. This needs longer texts...
```

Notes: Returns a value from 0 (not equal) to 100 (equal).

36.1.4 Spamsum(text as string, flags as Integer, blocksize as Integer) as string

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates a checksum for a given text.

Notes: You can specify for flags a combination of FlagsIgnoreWhitespace=1 and FlagsIgnoreHeaders=2.

Blocksize is linked to the length of the checksum. You can try values like 0, 3 or 90 and check what is best for you.

The text strings must be longer than just a few words as they are designed to take whole emails.

36.1.5 Properties

36.1.6 FlagsIgnoreHeaders as Integer

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The value used for the Spamsum call as flags.

Notes: Value is 2.

(Read only property)

36.1.7 FlagsIgnoreWhitespace as Integer

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The value used for the Spamsum call as flags.

Notes: Value is 1.

(Read only property)

Chapter 37

Special Folders

37.1 Globals

37.1.1 ALMLocationsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ALMLocationsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateALMLocationsFolder function.

37.1.2 ALMModulesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ALMModulesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateALMModulesFolder function.

37.1.3 ALMPreferencesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ALMPreferencesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateALMPreferencesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.4 AppearanceFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AppearanceFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAppearanceFolder function.

37.1.5 AppleExtrasFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AppleExtrasFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAppleExtrasFolder function.

37.1.6 AppleMenuFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AppleMenuFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAppleMenuFolder function.

37.1.7 AppleShareAuthenticationFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AppleShareAuthenticationFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAppleShareAuthenticationFolder function.

37.1.8 AppleshareAutomountServerAliasesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AppleshareAutomountServerAliasesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateAppleshareAutomountServerAliasesFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.9 AppleShareSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AppleShareSupportFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAppleShareSupportFolder function.

37.1.10 ApplicationsFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ApplicationsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateApplicationsFolder function.
On Windows, only the System Domain is supported.

37.1.11 ApplicationSupportFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
Var f as folderitem
```

```
f=preferencesFolderMBS(-32763)
if f=nil then // On Windows always nil
f=applicationSupportFolderMBS(-32763)
end if
```

```
msgBox f.NativePath
```

```
// example output:
```

```
// "Mac OS X:Users:cs:Library:Preferences:" on Mac OS X
// "Mac OS 9:Systemordner:Preferences" on a german Mac OS 9
// "C:\Dokumente und Einstellungen\Christian\Anwendungsdaten\" on a german Windows XP.
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateApplicationSupportFolder` function.

Xojo `ApplicationSupport` function points to `"/Library/Application Support"` while `ApplicationSupportFolderMBS` points to different locations on my test system depending on the domain code:

```
-32768 ->"/Library/Application Support"
-32767 ->"/Library/Application Support"
-32766 ->nil
-32765 ->"/Library/Application Support"
-32764 ->nil
-32763 ->"/Users/cs/Library/Application Support"
-32762 ->"/Volumes/Mac OS 9/Systemordner/Application Support"
0 ->"/Library/Application Support"
```

At last it is your decision to use the correct one!

37.1.12 AssistantsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AssistantsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAssistantsFolder function.

37.1.13 AudioAlertSoundsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioAlertSoundsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioAlertSoundsFolder function.

37.1.14 AudioComponentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioComponentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioComponentsFolder function.

37.1.15 AudioDigidesignFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioDigidesignFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateAudioDigidesignFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.16 AudioPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioPlugInsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioPlugInsFolder function.

37.1.17 AudioPresetsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioPresetsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioPresetsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.18 AudioSoundBanksFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioSoundBanksFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateAudioSoundBanksFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.19 AudioSoundsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioSoundsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioSoundsFolder function.

37.1.20 AudioSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioSupportFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioSupportFolder function.

37.1.21 AudioVSTFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AudioVSTFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAudioVSTFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.22 AutomatorWorkflowsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AutomatorWorkflowsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAutomatorWorkflowsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.23 AutosaveInformationFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = AutosaveInformationFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateAutosaveInformationFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.24 BootTimeStartupItemsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = BootTimeStartupItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateBootTimeStartupItemsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.25 CachedDataFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CachedDataFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateCachedDataFolder function.

37.1.26 CarbonLibraryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CarbonLibraryFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateCarbonLibraryFolder function.

37.1.27 ChewableItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ChewableItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateChewableItemsFolder function.

37.1.28 classicDesktopFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ClassicDesktopFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateClassicDesktopFolder function.

37.1.29 ClassicPreferencesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ClassicPreferencesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateClassicPreferencesFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.30 ColorPickersFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ColorPickersFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateColorPickersFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.31 ColorSyncCMMFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ColorSyncCMMFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateColorSyncCMMFolder function.

37.1.32 ColorSyncFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ColorSyncFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateColorSyncFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.33 ColorSyncProfilesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ColorSyncProfilesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateColorSyncProfilesFolder function.

37.1.34 ColorSyncScriptingFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:


```
const kUserDomain = -32763
Var f as folderitem = ColorSyncScriptingFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateColorSyncScriptingFolder function.

37.1.35 ComponentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ComponentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateComponentsFolder function.

37.1.36 CompositionsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CompositionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateCompositionsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.37 ContextualMenuItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ContextualMenuItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateContextualMenuItemsFolder` function.

37.1.38 `ControlPanelDisabledFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ControlPanelDisabledFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateControlPanelDisabledFolder` function.

37.1.39 ControlPanelFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ControlPanelFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateControlPanelFolder function.

37.1.40 ControlStripModulesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ControlStripModulesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateControlStripModulesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.41 CoreServicesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CoreServicesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateCoreServicesFolder function.

37.1.42 CreateALMLocationsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateALMLocationsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ALMLocationsFolder function.

37.1.43 CreateALMModulesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateALMModulesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ALMModulesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.44 CreateALMPreferencesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateALMPreferencesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ALMPreferencesFolder function.

37.1.45 CreateAppearanceFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAppearanceFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AppearanceFolder function.

37.1.46 CreateAppleExtrasFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAppleExtrasFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the AppleExtrasFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.47 CreateAppleMenuFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAppleMenuFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AppleMenuFolder function.

37.1.48 CreateAppleShareAuthenticationFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAppleShareAuthenticationFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AppleShareAuthenticationFolder function.

37.1.49 CreateAppleShareAutomountServerAliasesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAppleShareAutomountServerAliasesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the AppleShareAutomountServerAliasesFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.50 CreateAppleShareSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAppleShareSupportFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AppleShareSupportFolder function.

37.1.51 CreateApplicationsFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateApplicationsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ApplicationsFolder function.

On Windows, only the System Domain is supported.

37.1.52 CreateApplicationSupportFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateApplicationSupportFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ApplicationSupportFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Xojo ApplicationSupport function points to "/Library/Application Support" while ApplicationSupportFolderMBS points to different locations on my test system depending on the domain code:

```
-32768 ->"/Library/Application Support"
-32767 ->"/Library/Application Support"
-32766 ->nil
-32765 ->"/Library/Application Support"
-32764 ->nil
-32763 ->"/Users/cs/Library/Application Support"
-32762 ->"/Volumes/Mac OS 9/Systemordner/Application Support"
0 ->"/Library/Application Support"
```

At last it is your decision to use the correct one!

37.1.53 CreateAssistantsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAssistantsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the AssistantsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.54 CreateAudioAlertSoundsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioAlertSoundsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AudioAlertSoundsFolder function.

37.1.55 CreateAudioComponentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioComponentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AudioComponentsFolder function.

37.1.56 CreateAudioDigidesignFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioDigidesignFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the AudioDigidesignFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.57 CreateAudioPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioPlugInsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AudioPlugInsFolder function.

37.1.58 CreateAudioPresetsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioPresetsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AudioPresetsFolderMBS function.

37.1.59 CreateAudioSoundBanksFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioSoundBanksFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the AudioSoundBanksFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.60 CreateAudioSoundsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioSoundsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AudioSoundsFolder function.

37.1.61 CreateAudioSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioSupportFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AudioSupportFolder function.

37.1.62 CreateAudioVSTFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAudioVSTFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the AudioVSTFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.63 CreateAutomatorWorkflowsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAutomatorWorkflowsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AutomatorWorkflowsFolderMBS function.

37.1.64 CreateAutosaveInformationFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateAutosaveInformationFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the AutosaveInformationFolderMBS function.

37.1.65 CreateBootTimeStartupItemsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateBootTimeStartupItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the BootTimeStartupItemsFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.66 CreateCachedDataFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCachedDataFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CachedDataFolder function.

37.1.67 CreateCarbonLibraryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCarbonLibraryFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CarbonLibraryFolder function.

37.1.68 CreateChewableItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateChewableItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ChewableItemsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.69 CreateClassicDesktopFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateClassicDesktopFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ClassicDesktopFolder function.

37.1.70 CreateClassicPreferencesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateClassicPreferencesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ClassicPreferencesFolderMBS function.

37.1.71 CreateColorPickersFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateColorPickersFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ColorPickersFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.72 CreateColorSyncCMMFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateColorSyncCMMFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ColorSyncCMMFolder function.

37.1.73 CreateColorSyncFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateColorSyncFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ColorSyncFolder function.

37.1.74 CreateColorSyncProfilesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateColorSyncProfilesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ColorSyncProfilesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.75 CreateColorSyncScriptingFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateColorSyncScriptingFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ColorSyncScriptingFolder function.

37.1.76 CreateComponentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateComponentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ComponentsFolder function.

37.1.77 CreateCompositionsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCompositionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the CompositionsFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.78 CreateContextualMenuItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateContextualMenuItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ContextualMenuItemsFolder function.

37.1.79 CreateControlPanelDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateControlPanelDisabledFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ControlPanelDisabledFolder function.

37.1.80 CreateControlPanelFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateControlPanelFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ControlPanelFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.81 CreateControlStripModulesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateControlStripModulesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ControlStripModulesFolder function.

37.1.82 CreateCoreServicesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCoreServicesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CoreServicesFolder function.

37.1.83 CreateCurrentUserFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCurrentUserFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the CurrentUserFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.84 CreateCurrentUserRemoteFolderLocationFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCurrentUserRemoteFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CurrentUserRemoteFolder function.

37.1.85 CreateCurrentUserRemoteFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateCurrentUserRemoteFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CurrentUserRemoteFolder function.

37.1.86 CreateDesktopFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDesktopFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the SpecialFolder.Desktop function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.87 CreateDesktopPicturesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDesktopPicturesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DesktopPicturesFolder function.

37.1.88 CreateDeveloperApplicationsFolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** The developer application folder.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDeveloperApplicationsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DeveloperApplicationsFolderMBS function.

37.1.89 CreateDeveloperDocsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDeveloperDocsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the DeveloperDocsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.90 CreateDeveloperFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDeveloperFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DeveloperFolder function.

37.1.91 CreateDeveloperHelpFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDeveloperHelpFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DeveloperHelpFolder function.

37.1.92 CreateDictionariesFolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The folder for the dictionaries.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDictionariesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the DictionariesFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.93 CreateDirectoryServicesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDirectoryServicesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DirectoryServicesFolder function.

37.1.94 CreateDirectoryServicesPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDirectoryServicesPlugInsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DirectoryServicesPlugInsFolder function.

37.1.95 CreateDisplayExtensionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDisplayExtensionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the DisplayExtensionsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.96 CreateDocumentationFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDocumentationFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DocumentationFolder function.

37.1.97 CreateDocumentsFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDocumentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DocumentsFolder function.

On Windows only the user and the system domain are supported.

37.1.98 CreateDomainLibraryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDomainLibraryFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the DomainLibraryFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.99 CreateDomainTopLevelFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDomainTopLevelFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DomainTopLevelFolder function.

37.1.100 CreateDownloadsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateDownloadsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the DownloadsFolderMBS function.

37.1.101 CreateEditorsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateEditorsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the EditorsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.102 CreateExtensionDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateExtensionDisabledFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ExtensionDisabledFolder function.

37.1.103 CreateExtensionFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateExtensionFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ExtensionFolder function.

37.1.104 CreateFavoritesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFavoritesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the FavoritesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.105 CreateFileSystemSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFileSystemSupportFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the FileSystemSupportFolder function.

37.1.106 CreateFindByContentFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFindByContentFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the FindByContentFolder function.

37.1.107 CreateFindByContentIndexesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFindByContentIndexesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the FindByContentIndexesFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.108 CreateFindByContentPluginsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFindByContentPluginsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the FindByContentPluginsFolder function.

37.1.109 CreateFindSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFindSupportFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the FindSupportFolder function.

37.1.110 CreateFolderActionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFolderActionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the FolderActionsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.111 CreateFontCollectionsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFontCollectionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the FontCollectionsFolderMBS function.

37.1.112 CreateFontsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFontsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the FontsFolder function.

37.1.113 CreateFrameworksFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateFrameworksFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the FrameworksFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.114 CreateGenEditorsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateGenEditorsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the GenEditorsFolder function.

37.1.115 CreateHelpFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateHelpFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the HelpFolder function.

37.1.116 CreateiMovieFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateiMovieFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the iMovieFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.117 CreateiMoviePlugInsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateiMoviePlugInsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the iMoviePlugInsFolderMBS function.

37.1.118 CreateiMovieSoundEffectsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = iMovieSoundEffectsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateiMovieSoundEffectsFolderMBS function.

37.1.119 CreateIndexFilesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = IndexFilesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateIndexFilesFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.120 CreateInputManagersFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInputManagersFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the InputManagersFolderMBS function.

37.1.121 CreateInputMethodsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InputMethodsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateInputMethodsFolderMBS function.

37.1.122 CreateInstallerLogsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInstallerLogsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the InstallerLogsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.123 CreateInstallerReceiptsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInstallerReceiptsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the InstallerReceiptsFolder function.

37.1.124 CreateInternetFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInternetFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the InternetFolder function.

37.1.125 CreateInternetPlugInFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInternetPlugInFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the InternetPlugInFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.126 CreateInternetSearchSitesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInternetSearchSitesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the InternetSearchSitesFolder function.

37.1.127 CreateInternetSitesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateInternetSitesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the InternetSitesFolder function.

37.1.128 CreateISSDownloadsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateISSDownloadsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ISSDownloadsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.129 CreateKernelExtensionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateKernelExtensionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the KernelExtensionsFolder function.

37.1.130 CreateKeyboardLayoutsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = KeyboardLayoutsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateKeyboardLayoutsFolderMBS function.

37.1.131 CreateKeychainFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateKeychainFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the KeychainFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.132 CreateLauncherItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateLauncherItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the LauncherItemsFolder function.

37.1.133 CreateLibraryAssistantsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = LibraryAssistantsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateLibraryAssistantsFolderMBS function.

37.1.134 CreateLocalesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateLocalesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the LocalesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.135 CreateLogsFolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The log files folder.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateLogsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the LogsFolderMBS function.

37.1.136 CreateMacOSReadMesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateMacOSReadMesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the MacOSReadMesFolder function.

37.1.137 CreateMagicTemporaryItemsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MagicTemporaryItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateMagicTemporaryItemsFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.138 CreateManagedItemsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateManagedItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ManagedItemsFolderMBS function.

37.1.139 CreateMIDIDriversFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateMIDIDriversFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the MIDIDriversFolder function.

37.1.140 CreateModemScriptsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateModemScriptsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ModemScriptsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.141 CreateMovieDocumentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateMovieDocumentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the MovieDocumentsFolder function.

37.1.142 CreateMultiprocessingFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateMultiprocessingFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the MultiprocessingFolder function.

37.1.143 CreateMusicDocumentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateMusicDocumentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the MusicDocumentsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.144 CreateOpenDocEditorsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateOpenDocEditorsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the OpenDocEditorsFolder function.

37.1.145 CreateOpenDocFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateOpenDocFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the OpenDocFolder function.

37.1.146 CreateOpenDocLibrariesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateOpenDocLibrariesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the OpenDocLibrariesFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.147 CreateOpenDocShellPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateOpenDocShellPlugInsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the OpenDocShellPlugInsFolder function.

37.1.148 CreatePictureDocumentsFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePictureDocumentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PictureDocumentsFolder function.

As there is no such folder on Mac OS Classic, you always get nil there.

37.1.149 CreatePreferencePanefolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The preferences panes folder.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePreferencePanefolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the PreferencePanefolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.150 CreatePreferencesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePreferencesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PreferencesFolder function.

37.1.151 CreatePrinterDescriptionFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePrinterDescriptionFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PrinterDescriptionFolder function.

37.1.152 CreatePrinterDriverFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePrinterDriverFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the PrinterDriverFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.153 CreatePrintersFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePrintersFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PrintersFolder function.

37.1.154 CreatePrintingPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePrintingPlugInsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PrintingPlugInsFolder function.

37.1.155 CreatePrintMonitorDocsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePrintMonitorDocsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the PrintMonitorDocsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.156 CreatePrivateFrameworksFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePrivateFrameworksFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PrivateFrameworksFolder function.

37.1.157 CreatePublicFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreatePublicFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the PublicFolder function.

37.1.158 CreateQuickLookFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateQuickLookFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the QuickLookFolderMBS function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.159 CreateQuickTimeComponentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateQuickTimeComponentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the QuickTimeComponentsFolder function.

37.1.160 CreateQuickTimeExtensionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateQuickTimeExtensionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the QuickTimeExtensionsFolder function.

37.1.161 CreateRecentApplicationsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateRecentApplicationsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the RecentApplicationsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.162 CreateRecentDocumentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateRecentDocumentsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the RecentDocumentsFolder function.

37.1.163 CreateRecentServersFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateRecentServersFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the RecentServersFolder function.

37.1.164 CreateScriptingAdditionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateScriptingAdditionsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

Also take a look on the ScriptingAdditionsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.165 CreateScriptsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateScriptsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ScriptsFolder function.

37.1.166 CreateSharedLibrariesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSharedLibrariesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SharedLibrariesFolder function.

37.1.167 CreateSharedUserDataFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSharedUserDataFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SharedUserDataFolder function.

37.1.168 CreateShutdownFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateShutdownFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ShutdownFolder function.

37.1.169 CreateShutdownItemsDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateShutdownItemsDisabledFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ShutdownItemsDisabledFolder function.

37.1.170 CreateSoundSetsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSoundSetsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SoundSetsFolder function.

37.1.171 CreateSpeakableItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSpeakableItemsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SpeakableItemsFolder function.

37.1.172 CreateSpeechFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSpeechFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SpeechFolder function.

37.1.173 CreateSpotlightImportersFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpotlightImportersFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateSpotlightImportersFolderMBS` function.

37.1.174 `CreateSpotlightMetadataCacheFolderMBS(domain as Integer)` as `folderitem`

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpotlightMetadataCacheFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, `nil` is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateSpotlightMetadataCacheFolderMBS` function.

37.1.175 CreateSpotlightSavedSearchesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSpotlightSavedSearchesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SpotlightSavedSearchesFolderMBS function.

37.1.176 CreateStartupFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateStartupFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the StartupFolder function.

37.1.177 CreateStartupItemsDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateStartupItemsDisabledFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the StartupItemsDisabledFolder function.

37.1.178 CreateStationeryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateStationeryFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the StationeryFolder function.

37.1.179 CreateSystemControlPanelFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemControlPanelFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `SystemControlPanelFolder` function.

37.1.180 `CreateSystemDesktopFolderMBS(domain as Integer)` as `folderitem`

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemDesktopFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, `nil` is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `SystemDesktopFolder` function.

37.1.181 CreateSystemExtensionDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemExtensionDisabledFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SystemExtensionDisabledFolder function.

37.1.182 CreateSystemFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SystemFolder function.

37.1.183 CreateSystemPreferencesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemPreferencesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SystemPreferencesFolder function.

37.1.184 CreateSystemSoundsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemSoundsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SystemSoundsFolder function.

37.1.185 CreateSystemTrashFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateSystemTrashFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the SystemTrashFolder function.

37.1.186 CreateTemporaryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateTemporaryFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the TemporaryFolder function.

37.1.187 CreateTemporaryItemsInCacheDataFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateTemporaryItemsInCacheDataFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the TemporaryItemsInCacheDataFolderMBS function.

37.1.188 CreateTemporaryItemsInUserDomainFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateTemporaryItemsInUserDomainFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the TemporaryItemsInUserDomainFolderMBS function.

37.1.189 CreateTextEncodingsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateTextEncodingsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the TextEncodingsFolder function.

37.1.190 CreateThemesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateThemesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the ThemesFolder function.

37.1.191 CreateTrashFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateTrashFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the TrashFolder function.

37.1.192 CreateUsersFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateUsersFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the UsersFolder function.

37.1.193 CreateUserSpecificTmpFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateUserSpecificTmpFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the UserSpecificTmpFolder function.

37.1.194 CreateUtilitiesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateUtilitiesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the UtilitiesFolder function.

37.1.195 CreateVoicesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateVoicesFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the VoicesFolder function.

37.1.196 CreateVolumeRootFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateVolumeRootFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the VolumeRootFolder function.

37.1.197 CreateVolumeSettingsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateVolumeSettingsFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the VolumeSettingsFolder function.

37.1.198 CreateWhereToEmptyTrashFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CreateWhereToEmptyTrashFolderMBS(kUserDomain)
```

Notes: If this folder is supported by the operation system version, the folder is created and returned. If the folder could not be created or is not supported, nil is returned. If the folder already exists, it is just returned.

Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the WhereToEmptyTrashFolder function.

37.1.199 CurrentUserFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CurrentUserFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateCurrentUserFolder function.

37.1.200 CurrentUserRemoteFolderLocationFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CurrentUserRemoteFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateCurrentUserRemoteFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.201 CurrentUserRemoteFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = CurrentUserRemoteFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateCurrentUserRemoteFolder function.

37.1.202 DesktopFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:


```
const kUserDomain = -32763
Var f as folderitem = DesktopFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDesktopFolder function.

37.1.203 DesktopPicturesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DesktopPicturesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDesktopPicturesFolder function.

37.1.204 DeveloperApplicationsFolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The developer application folder.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DeveloperApplicationsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDeveloperApplicationsFolderMBS function.

37.1.205 DeveloperDocsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DeveloperDocsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateDeveloperDocsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.206 DeveloperFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DeveloperFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDeveloperFolder function.

37.1.207 DeveloperHelpFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DeveloperHelpFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDeveloperHelpFolder function.

37.1.208 DictionariesFolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The folder for the dictionaries.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DictionariesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateDictionariesFolderMBSfunction.

37.1.209 DirectoryServicesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DirectoryServicesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDirectoryServicesFolder function.

37.1.210 DirectoryServicesPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DirectoryServicesPlugInsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateDirectoryServicesPlugInsFolder` function.

37.1.211 `DisplayExtensionsFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DisplayExtensionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateDisplayExtensionsFolder` function.

37.1.212 `DocumentationFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DocumentationFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDocumentationFolder function.
On Windows only the user and the system domain are supported.

37.1.213 DocumentsFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DocumentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateDocumentsFolder function.

37.1.214 DomainLibraryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DomainLibraryFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateDomainLibraryFolder function.

37.1.215 DomainTopLevelFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DomainTopLevelFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateDomainTopLevelFolder` function.

37.1.216 DownloadsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = DownloadsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateDownloadsFolderMBS` function.

Blog Entries

- [Query URL from downloaded file](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)

- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.217 EditorsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = EditorsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateEditorsFolder function.

37.1.218 ExtensionDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ExtensionDisabledFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateExtensionDisabledFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.219 ExtensionFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ExtensionFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateExtensionFolder function.

37.1.220 FavoritesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FavoritesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateFavoritesFolder function.

37.1.221 FileSystemSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FileSystemSupportFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateFileSystemSupportFolder function.

37.1.222 FindByContentFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FindByContentFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateFindByContentFolder function.

37.1.223 FindByContentIndexesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FindByContentIndexesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateFindByContentIndexesFolderMBS` function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.224 FindByContentPluginsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FindByContentPluginsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateFindByContentPluginsFolder` function.

37.1.225 FindSupportFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FindSupportFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateFindSupportFolder function.

37.1.226 FolderActionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FolderActionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateFolderActionsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.227 FontCollectionsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FontCollectionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateFontCollectionsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.228 FontsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FontsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateFontsFolder function.

37.1.229 FrameworksFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = FrameworksFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateFrameworksFolder function.

37.1.230 GenEditorsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = GenEditorsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateGenEditorsFolder function.

37.1.231 HelpFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = HelpFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateHelpFolder function.

37.1.232 iMovieFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = iMovieFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateiMovieFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.233 iMoviePlugInsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = iMoviePlugInsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateiMoviePlugInsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.234 iMovieSoundEffectsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = iMovieSoundEffectsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateiMovieSoundEffectsFolderMBS` function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.235 IndexFilesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = IndexFilesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateIndexFilesFolderMBS` function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)

- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.236 InputManagersFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InputManagersFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateInputManagersFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.237 InputMethodsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InputMethodsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateInputMethodsFolderMBS` function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.238 `InstallerLogsFolderMBS(domain as Integer) as folderitem`

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InstallerLogsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateInstallerLogsFolder` function.

37.1.239 `InstallerReceiptsFolderMBS(domain as Integer)` as `folderitem`

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InstallerReceiptsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder `/System`.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateInstallerReceiptsFolder` function.

37.1.240 `InternetFolderMBS(domain as Integer)` as `folderitem`

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InternetFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateInternetFolder function.

37.1.241 InternetPlugInFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InternetPlugInFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateInternetPlugInFolder function.

37.1.242 InternetSearchSitesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InternetSearchSitesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateInternetSearchSitesFolder function.

37.1.243 InternetSitesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = InternetSitesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateInternetSitesFolder function.

37.1.244 ISSDownloadsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ISSDownloadsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateISSDownloadsFolder function.

37.1.245 KernelExtensionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = KernelExtensionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateKernelExtensionsFolder` function.

37.1.246 `KeyboardLayoutsFolderMBS(domain as Integer)` as `folderitem`

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = KeyboardLayoutsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateKeyboardLayoutsFolderMBS` function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.247 KeychainFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = KeychainFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateKeychainFolder function.

37.1.248 LauncherItemsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = LauncherItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateLauncherItemsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.249 LibraryAssistantsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = LibraryAssistantsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateLibraryAssistantsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.250 LocalesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = LocalesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateLocalesFolder function.

37.1.251 LogsFolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The log files folder.

Example:

```
const kUserDomain = -32763
Var f as folderitem = LogsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateLogsFolderMBS function.

37.1.252 MacOSReadMesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MacOSReadMesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateMacOSReadMesFolder function.

37.1.253 MagicTemporaryItemsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MagicTemporaryItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateMagicTemporaryItemsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.254 ManagedItemsFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ManagedItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateManagedItemsFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.255 MIDIDriversFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MIDIDriversFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateMIDIDriversFolder function.

37.1.256 ModemScriptsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ModemScriptsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateModemScriptsFolder function.

37.1.257 MovieDocumentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MovieDocumentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateMovieDocumentsFolder function.

37.1.258 MultiprocessingFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MultiprocessingFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateMultiprocessingFolder function.

37.1.259 MusicDocumentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = MusicDocumentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateMusicDocumentsFolder function.

37.1.260 OpenDocEditorsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = OpenDocEditorsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateOpenDocEditorsFolder function.

37.1.261 OpenDocFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = OpenDocFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateOpenDocFolder function.

37.1.262 OpenDocLibrariesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = OpenDocLibrariesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateOpenDocLibrariesFolder function.

37.1.263 OpenDocShellPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = OpenDocShellPlugInsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateOpenDocShellPlugInsFolder function.

37.1.264 PictureDocumentsFolderMBS(domain as Integer) as folderitem

Platforms: macOS, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PictureDocumentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreatePictureDocumentsFolder function.
As there is no such folder on Mac OS Classic, you always get nil there.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.265 PreferencePanefolderMBS(domain as Integer) as folderitem

Plugin Version: 3.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** The preferences panes folder.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PreferencePanefolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreatePreferencePanefolderMBS function.

37.1.266 PreferencesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

Var f as folderitem

```
f=preferencesFolderMBS(-32763)
if f=nil then // On Windows always nil
f=applicationsupportFolderMBS(-32763)
end if
```

msgBox f.NativePath

```
// example output:
// "Mac OS X:Users:cs:Library:Preferences:" on Mac OS X
// "Mac OS 9:Systemordner:Preferences" on a german Mac OS 9
// "C:\Dokumente und Einstellungen\Christian\Anwendungsdaten\" on a german Windows XP.
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreatePreferencesFolder function.

37.1.267 PrinterDescriptionFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PrinterDescriptionFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreatePrinterDescriptionFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.268 PrinterDriverFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PrinterDriverFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreatePrinterDriverFolder function.

37.1.269 PrintersFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PrintersFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreatePrintersFolder function.

37.1.270 PrintingPlugInsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PrintingPlugInsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreatePrintingPlugInsFolder function.

37.1.271 PrintMonitorDocsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PrintMonitorDocsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreatePrintMonitorDocsFolder function.

37.1.272 PrivateFrameworksFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PrivateFrameworksFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreatePrivateFrameworksFolder` function.

37.1.273 `PublicFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = PublicFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreatePublicFolder` function.

37.1.274 `QuickLookFolderMBS(domain as Integer)` as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = QuickLookFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateQuickLookFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.275 QuickTimeComponentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = QuickTimeComponentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateQuickTimeComponentsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.276 QuickTimeExtensionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = QuickTimeExtensionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateQuickTimeExtensionsFolder function.

37.1.277 RecentApplicationsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = RecentApplicationsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateRecentApplicationsFolder function.

37.1.278 RecentDocumentsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = RecentDocumentsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateRecentDocumentsFolder function.

37.1.279 RecentServersFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = RecentServersFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateRecentServersFolder function.

37.1.280 ScriptingAdditionsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ScriptingAdditionsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateScriptingAdditionsFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.281 ScriptsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ScriptsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateScriptsFolder function.

37.1.282 SharedLibrariesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SharedLibrariesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSharedLibrariesFolder function.

37.1.283 SharedUserDataFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SharedUserDataFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateSharedUserDataFolder function.

37.1.284 ShutdownFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ShutdownFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateShutdownFolder function.

37.1.285 ShutdownItemsDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ShutdownItemsDisabledFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateShutdownItemsDisabledFolder` function.

37.1.286 `SoundSetsFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SoundSetsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateSoundSetsFolder` function.

37.1.287 `SpeakableItemsFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpeakableItemsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSpeakableItemsFolder function.

37.1.288 SpeechFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpeechFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateSpeechFolder function.

37.1.289 SpotlightImportersFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpotlightImportersFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSpotlightImportersFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.290 SpotlightMetadataCacheFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpotlightMetadataCacheFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSpotlightMetadataCacheFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.291 SpotlightSavedSearchesFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SpotlightSavedSearchesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateSpotlightSavedSearchesFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.292 StartupFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = StartupFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateStartupFolder function.

37.1.293 StartupItemsDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = StartupItemsDisabledFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateStartupItemsDisabledFolder function.

37.1.294 StationeryFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = StationeryFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateStationeryFolder function.

37.1.295 SystemControlPanelFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemControlPanelFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSystemControlPanelFolder function.

37.1.296 SystemDesktopFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemDesktopFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateSystemDesktopFolder function.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

37.1.297 SystemExtensionDisabledFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemExtensionDisabledFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSystemExtensionDisabledFolder function.

37.1.298 SystemFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSystemFolder function.

37.1.299 SystemPreferencesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemPreferencesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateSystemPreferencesFolder function.

37.1.300 SystemSoundsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemSoundsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateSystemSoundsFolder function.

37.1.301 SystemTrashFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = SystemTrashFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateSystemTrashFolder` function.

37.1.302 `TemporaryFolderMBS(domain as Integer)` as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = TemporaryFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateTemporaryFolder` function.

37.1.303 `TemporaryItemsInCacheDataFolderMBS(domain as Integer)` as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = TemporaryItemsInCacheDataFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateTemporaryItemsInCacheDataFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.304 TemporaryItemsInUserDomainFolderMBS(domain as Integer) as folderitem

Plugin Version: 10.4, Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = TemporaryItemsInUserDomainFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateTemporaryItemsInUserDomainFolderMBS function.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

- [MBS REALbasic Plugins, version 10.4pr5](#)

37.1.305 TextEncodingsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = TextEncodingsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateTextEncodingsFolder function.

37.1.306 ThemesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = ThemesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateThemesFolder function.

37.1.307 TrashFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = TrashFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateTrashFolder function.

37.1.308 UsersFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = UsersFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateUsersFolder function.

37.1.309 UserSpecificTmpFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = UserSpecificTmpFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateUserSpecificTmpFolder` function.

37.1.310 UtilitiesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = UtilitiesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateUtilitiesFolder` function.

37.1.311 VoicesFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = VoicesFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateVoicesFolder function.

37.1.312 VolumeRootFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = VolumeRootFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

Also take a look on the CreateVolumeRootFolder function.

37.1.313 VolumeSettingsFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = VolumeSettingsFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!
The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the CreateVolumeSettingsFolder function.

37.1.314 WhereToEmptyTrashFolderMBS(domain as Integer) as folderitem

Platform: macOS, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use SpecialFolder module instead. **Function:** Returns a folderitem to this folder if it exists on that system.

Example:

```
const kUserDomain = -32763
Var f as folderitem = WhereToEmptyTrashFolderMBS(kUserDomain)
```

Notes: Note that not every folder exists on every Mac OS Version!

The codes for domain are (The Demo includes a module for these):

- 32768 On System Disk
- 32767 On Appropriate Disk
- 32766 System Domain, on Mac OS X mostly inside the folder /System.
- 32765 Local Domain (on Netbooting for example)
- 32764 Network Domain (on Netbooting for example)
- 32763 User Domain, on Mac OS X mostly inside the users folder.
- 32762 Classic Domain, the current used Classic System folder.

Also take a look on the `CreateWhereToEmptyTrashFolder` function.

37.1.315 `WindowsBurnAreaFolderMBS` as `folderitem`

Plugin Version: 8.5, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns the folder which Windows uses to store temporary data for burning a CD/DVD.

Notes: Returns nil on any error.

37.1.316 `WindowsFolderMBS` as `folderitem`

Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
Var f as folderitem
f=WindowsFolderMBS
```

37.1.317 `WindowsSystemFolderMBS` as `folderitem`

Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use `SpecialFolder` module instead. **Function:** Returns a `folderitem` to this folder if it exists on that system.

Example:

```
Var f as folderitem
```

f=WindowsSystemFolderMBS

Chapter 38

String

38.1 Globals

38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds bytes or byte ranges in a string.

Example:

```
// search for question mark or uppercase letters A to Z
MsgBox str(InStrByteRangeMBS("Hello?", "?", 65, 90)) // shows 1 as H is found.
```

```
MsgBox str(InStrByteRangeMBS("hello?", "?", 65, 90)) // shows 6 as ? is found.
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The optional find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

The StartValue/EndValue parameters define ranges of values which you consider to match.

Returns 0 if nothing is found or the position of the found character.

See also:

- 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090

- 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093

38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds bytes or byte ranges in a string.

Example:

```
// search for question mark or uppercase letters A to Z or numbers from 0 to 9
MsgBox str(InStrByteRangeMBS("Hello?", "?", 65, 90, 48, 57)) // shows 1 as H is found.
MsgBox str(InStrByteRangeMBS("hello?", "?", 65, 90, 48, 57)) // shows 6 as ? is found
MsgBox str(InStrByteRangeMBS("Hello 123", "?", 65, 90, 48, 57)) // shows 1 as H is found.
MsgBox str(InStrByteRangeMBS("hello 123", "?", 65, 90, 48, 57)) // shows 7 as 1 is found.
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The optional find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

The StartValue/EndValue parameters define ranges of values which you consider to match.

Returns 0 if nothing is found or the position of the found character.

See also:

- 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091

- 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer
1092
- 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer
1093
- 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer
1093

38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds bytes or byte ranges in a string.

Example:

```
// search for question mark, low byte values, high byte values or numbers from 0 to 9
MsgBox str(InStrByteRangeMBS("Hello?", "?", 0, 31, 127, 255, 48, 57)) // shows 6 as ? is found.
MsgBox str(InStrByteRangeMBS("123", "?", 0, 31, 127, 255, 48, 57)) // shows 1 as 1 is found.
MsgBox str(InStrByteRangeMBS(EndOfLine.Windows, "?", 0, 31, 127, 255, 48, 57)) // shows 1 as chr(13)
is found.
MsgBox str(InStrByteRangeMBS("Hello", "?", 0, 31, 127, 255, 48, 57)) // shows 0 as nothing is found.
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The optional find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

The StartValue/EndValue parameters define ranges of values which you consider to match.

Returns 0 if nothing is found or the position of the found character.

See also:

- 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer
1089
- 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer
1090
- 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer
1092
- 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer
1093

- 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093

38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds bytes or byte ranges in a string.

Example:

```
// search for uppercase letters A to Z
MsgBox str(InStrByteRangeMBS("Hello?", 65, 90)) // shows 1 as H is found.
```

```
MsgBox str(InStrByteRangeMBS("hello?", 65, 90)) // shows 0 as nothing is found.
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The optional find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

The StartValue/EndValue parameters define ranges of values which you consider to match.

Returns 0 if nothing is found or the position of the found character.

See also:

- 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093
- 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093

38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds bytes or byte ranges in a string.

Example:

```
// search for uppercase letters A to Z or numbers from 0 to 9
MsgBox str(InStrByteRangeMBS("Hello?", 65, 90, 48, 57)) // shows 1 as H is found.
MsgBox str(InStrByteRangeMBS("hello?", 65, 90, 48, 57)) // shows 0 as nothing is found.
MsgBox str(InStrByteRangeMBS("Hello 123", 65, 90, 48, 57)) // shows 1 as H is found.
MsgBox str(InStrByteRangeMBS("hello 123", 65, 90, 48, 57)) // shows 7 as 1 is found.
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The optional find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

The StartValue/EndValue parameters define ranges of values which you consider to match.

Returns 0 if nothing is found or the position of the found character.

See also:

- 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- 38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1093

38.1.6 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds bytes or byte ranges in a string.

Example:

```
// search for low byte values, high byte values or numbers from 0 to 9
MsgBox str(InStrByteRangeMBS("Hello?", 0, 31, 127, 255, 48, 57 )) // shows 0 as nothing is found.
MsgBox str(InStrByteRangeMBS("123", 0, 31, 127, 255, 48, 57 )) // shows 1 as 1 is found.
MsgBox str(InStrByteRangeMBS(EndOfLine.Windows, 0, 31, 127, 255, 48, 57 )) // shows 1 as chr(13) is
found.
MsgBox str(InStrByteRangeMBS("Hello", 0, 31, 127, 255, 48, 57 )) // shows 0 as nothing is found.
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The optional find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

The StartValue/EndValue parameters define ranges of values which you consider to match.

Returns 0 if nothing is found or the position of the found character.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr11](#)

See also:

- 38.1.1 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer) as Integer 1089
- 38.1.2 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1090
- 38.1.3 InStrByteRangeMBS(target as string, find as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer, StartValue3 as Integer, EndValue3 as Integer) as Integer 1091
- 38.1.4 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer) as Integer 1092
- 38.1.5 InStrByteRangeMBS(target as string, StartValue as Integer, EndValue as Integer, StartValue2 as Integer, EndValue2 as Integer) as Integer 1093

38.1.7 InStrBytesMBS(target as string, find as string) as Integer

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the first position of one of the bytes given in the find string in the target string.

Example:

```
MsgBox str(InStrBytesMBS("Hello", "e")) // shows 2, as e is found
MsgBox str(InStrBytesMBS("Hello", "abcd")) // shows 0, as nothing is found
MsgBox str(InStrBytesMBS("Hello World", "abcd")) // shows 11, as d is found
```

Notes: You may want to make sure target and find are both in the same 8bit encoding (UTF-8, MacRoman, Windows ANSI, Latin 1, ASCII, etc. but not UTF16/UTF32).

The find string defines the bytes to search. This gives you a way to see whether one of those bytes is inside or to find the first byte matching a certain value.

Returns 0 if nothing is found or the position of the found character.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr11](#)

38.1.8 EncodingNameMBS(extends Text as string) as string

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the text encoding name for this string.

Example:

```
Var t as string = "Hello World"
```

```
MsgBox "InternetName: "+t.Encoding.InternetName
MsgBox "InternetNameMBS: "+t.Encoding.InternetNameMBS
MsgBox "EncodingNameMBS: "+t.EncodingNameMBS
```

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 13.5pr6](#)

38.1.9 RemoveAccentsMBS(text as string, IgnoreCase as boolean = false) as string

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes accents from text.

Example:

```
Var s as string = "L'√©l√@ve pr√©f√©r√©"
Var a as string = RemoveAccentsMBS(s)
```

```

Var b as string = RemoveAccentsMBS(s, true)
// check values in debugger
Var x As String = RemoveAccentsMBS("“ ,Äò ,Äô ,Äœ \¶–® √Ä √Å √Ç √É √Ñ √Ö √Ü √á √à √â √ã √ä √å √ç
√é √è √ê √ë √í √î √ï √î √ñ √ò √ó √ô √õ √ú √ù √û √ü √† √° √c √£ √§ √• √¶ √ß √® √© √™ √’ √” √√Æ √Ø √∞ √±
√√ √¥ √µ √√ √√ √√ √ª √º √Ω √æ √ø −º −Ω −æ ,Öì ,Öï ,Öî ,Öñ ,Öó ,Öò ,Öö ,Öè ,Öõ ,Öú ,Öù ,Öë ,Öí ,Üâ”)

```

Break

Notes: Optionally also ignores case and outputs text in capital letters.
The plugin has a long replacement list of unicode code points to do this.

Useful as a preparation to convert to ASCII text, so all accents are removed before conversion.

Blog Entries

- [MBS Xojo Plugins, version 24.5pr1](#)
- [MBS Xojo Plugins, version 22.2pr5](#)
- [MBS SQLite Extension in version 1.5](#)
- [MBS Xojo Plugins, version 18.5pr2](#)
- [MBS Xojo Plugins, version 18.2pr1](#)
- [MBS Real Studio Plugins, version 12.2pr6](#)
- [MBS Real Studio Plugins, version 12.0pr4](#)
- [\[ANN \] MonkeyBread Software starts plug-in development for REAL Server](#)

Xojo Developer Magazine

- [17.4, page 8: News](#)

38.1.10 SplitCommaSeparatedValuesMBS(text as string, delimiter as string = ",", quote as string = "") as string()

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Splits a string with CSV values.

Example:

```

// 1. read a CSV file:

Var file as FolderItem = SpecialFolder.Desktop.Child("test.csv")
Var t as TextInputStream = TextInputStream.open(file)

// use right encoding for your file!

```



```
t.Encoding = encodings.MacRoman

while not t.EOF
  Var line as string = t.ReadLine
  Var items() as string = SplitCommaSeparatedValuesMBS(line)

  // process items array
wend

// 2. compare with split:

Var test as string = "Hello,""World,test""end"

Var a() as string = split(test, ",")
Var b() as string = SplitCommaSeparatedValuesMBS(test, ",")

MsgBox Join(a,EndOfLine)+EndOfLine+EndOfLine+Join(b,EndOfLine)
```

Notes: This function is better for CSV data than the Split function as it handles quoted text right. For 17.5 we rewrote this function. Now auto detects the delimiter (comma or semicolon) if none passed. Default quote character is " if nothing is passed. Please only one character for delimiter.

Version 18.0 can detect tab character, too.

Blog Entries

- [MBS Xojo Plugins, version 22.5pr1](#)
- [MBS Xojo Plugins, version 22.4pr1](#)
- [MBS Xojo Plugins, version 22.2pr1](#)
- [MBS Xojo Plugins, version 19.3pr1](#)
- [MBS Xojo Plugins, version 18.6pr3](#)
- [MBS Xojo Plugins, version 18.5pr4](#)
- [MBS Xojo Plugins, version 18.0pr8](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.5](#)
- [MBS Xojo Plugins, version 17.5pr4](#)
- [MBS Xojo / Real Studio Plugins, version 16.5pr9](#)

Videos

- [Presentation from Munich conference about MBS Plugins.](#)

Xojo Developer Magazine

- [16.1, page 10: News](#)

38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns string as UTF-8 and replacing invalid UTF-8 sequences with placeholder.

Notes: This function is more save than simply DefineEncoding as it makes sure the returned text actually is valid UTF-8.

You can replace missing characters with empty text, question mark or any other symbol.

This function is overloaded, so you can directly pass in string, memoryblock or ptr+size.

Checks byte sequence with up to 4 byte long sequences. Does not verify whether code points are valid.

See also:

- [38.1.12 CheckUTF8MBS\(data as string, Placeholder as string\) as string](#) 1098
- [38.1.13 CheckUTF8MBS\(mem as MemoryBlock, Placeholder as string\) as string](#) 1099

38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns string as UTF-8 and replacing invalid UTF-8 sequences with placeholder.

Example:

```
// some random bytes
Var t as string = RandomBytesStringMBS(20, false)

// and some text
t = t + " √§√√° üòÄ"

// checked
Var s as string = CheckUTF8MBS(t, "üòÇ")
```

MsgBox s

Notes: This function is more save than simply DefineEncoding as it makes sure the returned text actually is valid UTF-8.

You can replace missing characters with empty text, question mark or any other symbol.

This function is overloaded, so you can directly pass in string, memoryblock or ptr+size.

Checks byte sequence with up to 4 byte long sequences. Does not verify whether code points are valid.

See also:

38.1. GLOBALS	1099
• 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string	1098
• 38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string	1099

38.1.13 CheckUTF8MBS(mem as MemoryBlock, Placeholder as string) as string

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns string as UTF-8 and replacing invalid UTF-8 sequences with placeholder.

Example:

```
// some random bytes
Var t as string = RandomBytesStringMBS(20, false)

// and some text
t = t + " √§√√° üòÄ"

// checked in memoryblock
Var m as MemoryBlock = t
Var s as string = CheckUTF8MBS(m, "üòÇ")
```

MsgBox s

Notes: This function is more save than simply DefineEncoding as it makes sure the returned text actually is valid UTF-8.

You can replace missing characters with empty text, question mark or any other symbol.

This function is overloaded, so you can directly pass in string, memoryblock or ptr+size.

Checks byte sequence with up to 4 byte long sequences. Does not verify whether code points are valid.

Blog Entries

- [MBS Xojo Plugins, version 21.1pr1](#)
- [MBS Xojo plug-ins in version 16.0](#)
- [MBS Xojo / Real Studio Plugins, version 16.0pr7](#)
- [CheckUTF8MBS function](#)

Xojo Developer Magazine

- [14.2, page 10: News](#)

See also:

- | | |
|---|------|
| • 38.1.11 CheckUTF8MBS(data as ptr, size as Integer, Placeholder as string) as string | 1098 |
| • 38.1.12 CheckUTF8MBS(data as string, Placeholder as string) as string | 1098 |

38.1.14 ClearStringContentMBS(s as String) as Boolean

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears string content.

Example:

```
// this is a constant string
Var c As String = "Hello"

// this is an allocated string
Var a As String = c + "World"

// try to clear both
Call ClearStringContentMBS c
Call ClearStringContentMBS a
```

```
MsgBox "const: "+c+" "+Str(Len(c))+EndOfLine+"allocated: "+a+" "+Str(Len(a))
```

Notes: Overwrites content of string with zeros.

We check if this is a constant string and for those do nothing.

Returns true if we zeroed content, otherwise false.

This function can help to clear memory for passwords stored in strings.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 19.5](#)
- [MBS Xojo Plugins, version 19.5pr6](#)
- [MBS Real Studio Plugins, version 11.2pr8](#)
- [MonkeyBread Software Releases the MBS Plugins 8.2](#)

38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Concat the given strings with binary encoding.

Example:

```
Var a as string = "Hello"
Var b as string = "World"
Var c as string = ConcatBinaryStringsMBS(a,b)
Var d as string = ConvertEncoding(b, encodings.UTF16)
Var e as string = ConcatBinaryStringsMBS(a,d)
```

MsgBox "ConcatBinaryStringsMBS"+EndOfLine+EndOfLine+c+EndOfLine+EncodingToHexMBS(c)+EndOfLine+EndOfLine+e+EndOfLine+EncodingToHexMBS(e)

Notes: If RB concats strings the encoding is changed before concatenating to match the other strings.
See also:

- 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string 1101
- 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string 1101
- 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string 1102
- 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string 1102

38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Concat the given strings with binary encoding.

Notes: If RB concats strings the encoding is changed before concatenating to match the other strings.
See also:

- 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string 1100
- 38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string 1101
- 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string 1102
- 38.1.19 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string 1102

38.1.17 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Concat the given strings with binary encoding.

Notes: If RB concats strings the encoding is changed before concatenating to match the other strings.
See also:

- 38.1.15 ConcatBinaryStringsMBS(a as string, b as string) as string 1100
- 38.1.16 ConcatBinaryStringsMBS(a as string, b as string, c as string) as string 1101
- 38.1.18 ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string 1102

- 38.1.19 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string` 1102

38.1.18 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string`

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Concats the given strings with binary encoding.

Notes: If RB concat strings the encoding is changed before concatenating to match the other strings.

See also:

- 38.1.15 `ConcatBinaryStringsMBS(a as string, b as string) as string` 1100
- 38.1.16 `ConcatBinaryStringsMBS(a as string, b as string, c as string) as string` 1101
- 38.1.17 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string` 1101
- 38.1.19 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string` 1102

38.1.19 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string, f as string) as string`

Plugin Version: 6.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Concats the given strings with binary encoding.

Notes: If RB concat strings the encoding is changed before concatenating to match the other strings.

Blog Entries

- [MBS Xojo Plugins, version 19.1pr7](#)

See also:

- 38.1.15 `ConcatBinaryStringsMBS(a as string, b as string) as string` 1100
- 38.1.16 `ConcatBinaryStringsMBS(a as string, b as string, c as string) as string` 1101
- 38.1.17 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string) as string` 1101
- 38.1.18 `ConcatBinaryStringsMBS(a as string, b as string, c as string, d as string, e as string) as string` 1102

38.1.20 `ContainsWholeWordMBS(Text as String, Word as String) as boolean`

Plugin Version: 22.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks whether text contains whole word.

Example:

```
// let's try a few test cases

If ContainsWholeWordMBS("Hello World", "Hello") Then
// okay
Else
Break
End If

If ContainsWholeWordMBS("Hello World", "World") Then
// okay
Else
Break
End If

If ContainsWholeWordMBS("Hello World", "ello") Then
Break
Else
// okay
End If

If ContainsWholeWordMBS("Hello World", "Worl") Then
Break
Else
// okay
End If

If ContainsWholeWordMBS("Hello World", "orld") Then
Break
Else
// okay
End If

If ContainsWholeWordMBS("Hello World!", "orld") Then
Break
Else
// okay
End If
```

Notes: This function is optimized for speed and should only be used with ASCII as it handles all unicode characters as word characters.

Returns true if whole word is found inside text.

Returns false if the word was either not found or is contained, but has other letters before/after.

This function should be faster than doing a RegEx with "\b" + word "\b", but the RegEx may handle

unicode word delimiters better.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [News from the MBS Xojo Plugins Version 22.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 22.1](#)
- [MBS Xojo Plugins, version 22.1pr2](#)

Xojo Developer Magazine

- [20.3, page 10: News](#)

38.1.21 CountOccurrencesMBS(s as string, find as string) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Counts the occurrences of a string.

Example:

```
Var s as string
```

```
s="111110001110101010111"
```

```
MsgBox str(CountOccurrencesMBS(s,"1")) // 14
```

```
MsgBox str(CountOccurrencesMBS(s,"0")) // 7
```

```
MsgBox str(len(s)) // 21
```

Notes: Returns the number of occurrences of a substring in a string.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.2pr5](#)
- [MonkeyBread Software Releases the MBS Plugins 8.2](#)

38.1.22 CreateStringMBS(Length as Integer, Content as String) as string

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a string based on a pattern.

Example:

```
Var c as string = CreateStringMBS(20, "Hello")
```

```
Var t as string = CreateStringMBS(10, chr(255))
```


[Break](#)

Notes: The content string is repeated until size is matched and returned. The string returned has the same encoding set as the content string.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.4pr5](#)

38.1.23 DecodingFromHTMLMBS(s as string) as string

Plugin Version: 2.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with HTML escaped characters.

Example:

```
Var html as string = "<p><B>Gr&uuml;&szlig;e</B></P>"
Var htmltext as string = RemoveHTMLTagsMBS(html)
Var text as string = DecodingFromHTMLMBS(htmltext)
```

MsgBox text // shows: Gr^öüe

Notes: The source string is converted to a native string if it is not already in ASCII. Then for every character the functions looks for the unescaped character code and returns a normal unicode string.

e.g. "München" -> "M^ünchen"

This functions uses Xojo unicode strings, so you may need to convert back to a NativeString before saving the string to a file. (using e.g. ConvertEncoding(string, encodings.UTF8))

May return "" on low memory conditions.

Strings and encoding work only perfectly for RB 4.5 or newer.

Speed optimized in version 2.8 to be a hundred times faster.

Added Linux support in v5.1.

The result of this function is unpredictable on bad input strings.

(e.g. no encoding, or encoding does not match the content of the string)

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 15.0pr7](#)

38.1.24 DecodingFromMySQLMBS(s as string) as string

Plugin Version: 11.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string from MySQL.

Example:

```
MsgBox DecodingFromMySQLMBS("test\\test\%2.doc")
```

Notes: Replaces all the escapes in a mysql string like (\0, \t or \\) to their binary replacement.

38.1.25 DecodingFromQuotedPrintableMBS(s as string) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with quoted printable encoding.

Example:

```
msgbox DecodingFromQuotedPrintableMBS("Hi, =A1=92")
```

Notes: The decoded string is marked to have an ASCII encoding, but you may need to set the encoding to something like ISO. Then you need a text converter to make something printable from it.

May return "" on low memory conditions.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.3pr9](#)

38.1.26 DecodingFromURLMBS(s as string) as string

Plugin Version: 2.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with URL escaped characters.

Example:

```
Var s as string
```

```
Var t as string
```

```
Var u as string
```

```
t="Hello World √§√√°"
```

```
s= EncodingToURLMBS(t,0)
```

```
u=DecodingFromURLMBS(s,0)
```

```

SetEncodingOfStringMBS u,GetEncodingOfStringMBS(t) // restore encoding

// Hello World √§√√°
MsgBox t
// Hello%20World%20%C3%A4%C3%B6%C3%BC
MsgBox s
// Hello World √§√√°
MsgBox u

```

Notes: Decodes an URL encoded ASCII string. The string returned is marked as a binary string (without encoding). You need to set the encoding to whatever the original was (ISO-9660 or UTF8 for example).

e.g. "Wie%20geht's%3F" ->"Wie geht's?"

May return "" on low memory conditions.
 Strings and encoding work only perfectly for RB 4.5 or newer.
 Added Linux support in v5.1.
 See also:

- 38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string

1107

38.1.27 DecodingFromURLMBS(s as string, options as Integer) as string

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with URL escaped characters.

Example:

```

Var s as string
Var t as string
Var u as string

```

```
t="Hello World √§√√°"
```

```
s= EncodingToURLMBS(t,1)
```

```
u=DecodingFromURLMBS(s,1)
```

```

SetEncodingOfStringMBS u,GetEncodingOfStringMBS(t) // restore encoding

// Hello World √§√√°
MsgBox t
// Hello+World+%C3%A4%C3%B6%C3%BC
MsgBox s
// Hello World √§√√°
MsgBox u

```

Notes: Decodes an URL encoded ASCII string. The string returned is marked as a binary string (without encoding). You need to set the encoding to whatever the original was (ISO-9660 or UTF8 for example).

e.g. "Wie%20geht's%3F" ->"Wie geht's?"

May return "" on low memory conditions.
Strings and encoding work only perfectly for RB 4.5 or newer.

Pass 1 for the options parameter to get PHP/Perl compatible output (+ instead of spaces and %20 for spaces).

Added Linux support in v5.1.

See also:

- 38.1.26 `DecodingFromURLMBS(s as string) as string` 1106

38.1.28 `DecodingFromXMLMBS(s as string) as string`

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with XML escaped characters.

Notes: see `DecodingFromHTMLMBS`.

38.1.29 `DetectUnicodeMarkersMBS(s as string) as Integer`

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tries to get the unicode marker from the string.

Example:

```
const EncodingUnknown=0
const EncodingUTF8=1
const EncodingUTF16be=3
const EncodingUTF16le=4
const EncodingUTF32be=6
const EncodingUTF32le=7
```

```
Var text as string = "Hello World" // this text does not have a marker..
```

```
msgbox str(DetectUnicodeMarkersMBS(text))
```

Notes: For some UTF16 LittleEndian strings this function may return UTF32.

38.1.30 EncodeEmailSubjectMBS(s as string) as string

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function:

Encodes an email subject.

Text is encoded with UTF-8 Quoted Printable encoding.

If text does not need to be encoded (pure ASCII), we return the input text.

Example:

```
MsgBox EncodeEmailSubjectMBS("Hello World")+EndOfLine+EncodeEmailSubjectMBS("Test √$√√°")
```

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.3pr3](#)

38.1.31 EncodingToHTMLMBS(s as string, options as Integer = 0) as string

Plugin Version: 2.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string with HTML escaped characters.

Example:

```
Var f as folderItem
```

```
Var b as binaryStream
```

```
f=getsaveFolderItem("text/html","new.html")
if f<>nil then
b=f.createBinaryFile("text/html")
if b<>nil then
b.write nativeStringMBS(EncodingToHTMLMBS(editfield1.text))
b.close
end if
end if
```

Notes: The source string is converted to unicode if it is not already in unicode. Then for every character the functions looks for the escaped character code and returns a HTML encoded string.

e.g. "M√°nchen" ->"München"

This functions uses Xojo unicode strings, so you may need to convert back to a NativeString before saveing the string to a file.

Return characters (chr(10) and chr(13)) are not converted to
codes.

May return "" on low memory conditions.

The result of this function is unpredictable on bad input strings.
(e.g. no encoding, or encoding does not match the content of the string)

Options can be 1 to not encode ASCII values <128, so quotes, <, >and & are not encoded.

Blog Entries

- [MBS Xojo Plugins, version 23.5pr2](#)
- [Smileys and UTF32](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr9](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr7](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr7](#)
- [MBS Real Studio Plugins, version 11.2pr2](#)

38.1.32 EncodingToQuotedPrintableMBS(s as string, LineLen as Integer = 72) as string

Plugin Version: 5.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string with quoted printable characters.

Example:

```

Var a as string = "Hello √§√/°!"
Var b as string = EncodingToQuotedPrintableMBS(a)
Var c as string = ConvertEncoding(a, encodings.ISOLatin1)
Var d as string = EncodingToQuotedPrintableMBS(c)
Var e as string = EncodeQuotedPrintable(a)
Var f as string = EncodeQuotedPrintable(c)

```

```
MsgBox b+EndOfLine+d+EndOfLine+e+EndOfLine+f
```

Notes: Line wrap is per default at 72 characters but you can pass some other positive value here.
Quoted printable encoded strings have some ISO encoding as base, you need to pass a valid ISO string to this function to get the correct result.

May return "" on low memory conditions.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.3pr9](#)

38.1.33 EncodingToURLMBS(s as string) as string

Plugin Version: 2.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string with URL escaped characters.

Example:

```
msgbox EncodingToURLMBS("Wie geht's?")
```

Notes: Encodings a string for an URL. Use with UTF8 or ISO-9660 encoded strings. This function does not work correctly with UTF16 strings.

e.g. "Wie geht's?" -> "Wie%20geht's%3F"

May return "" on low memory conditions.

Strings and encoding work only perfectly for RB 4.5 or newer.

Added Linux support in v5.1.

See also:

- 38.1.34 EncodingToURLMBS(s as string, options as Integer) as string

1111

38.1.34 EncodingToURLMBS(s as string, options as Integer) as string

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string with URL escaped characters.

Example:

```
Var s as string = "20101210 1244 - Sky Cinema +24 - Wen die Geister lieben"
```

```
MsgBox EncodingToURLMBS(s)
```

```
// gives: 20101210%201244%20-%20Sky%20Cinema%20+24%20-%20Wen%20die%20Geister%20lieben
```

```
MsgBox EncodingToURLMBS(s,1)
```

```
// gives: 20101210+1244+-+Sky+Cinema+%2B24+-+Wen+die+Geister+lieben
```

```
MsgBox EncodingToURLMBS(s,2)
```

```
// gives: 20101210%201244%20-%20Sky%20Cinema%20%2b24%20-%20Wen%20die%20Geister%20lieben
```

Notes: Encodings a string for an URL. Use with UTF8 or ISO-9660 encoded strings. This function does not work correctly with UTF16 strings.

e.g. "Wie geht's?" ->"Wie%20geht's%3F"

May return "" on low memory conditions.
Strings and encoding work only perfectly for RB 4.5 or newer.

Pass 1 for the options parameter to get PHP/Perl compatible output (+ instead of spaces and %20 for spaces).

Pass 2 for the options parameter to get plus to %2B and space to %20. Added in plugin version 10.6.

Blog Entries

- [MBS REALbasic Plugins, version 10.6pr8](#)

See also:

- 38.1.33 `EncodingToURLMBS(s as string) as string` 1111

38.1.35 `EncodingToXMLMBS(s as string, options as Integer = 0) as string`

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string with XML escaped characters.

Notes: see `EncodingToHTMLMBS`.

38.1.36 `GetStringFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string()`

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Looks for strings within a data block.

Notes: Looks over the bytes in the data block to see if there are useful byte sequences which could be text. Returns an array with all text fragments found.

This function is UTF8 aware and will work well for UTF-8 encoded text fragments.

See also:

- 38.1.37 `GetStringFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string()` 1113
- 38.1.38 `GetStringFromDataMBS(data as String, MinLength as Integer = 0) as string()` 1113

38.1.37 GetStringsFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string()

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Looks for strings within a data block.

Notes: Looks over the bytes in the data block to see if there are useful byte sequences which could be text. Returns an array with all text fragments found.

This function is UTF8 aware and will work well for UTF-8 encoded text fragments.

See also:

- 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string() 1112
- 38.1.38 GetStringsFromDataMBS(data as String, MinLength as Integer = 0) as string() 1113

38.1.38 GetStringsFromDataMBS(data as String, MinLength as Integer = 0) as string()

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Looks for strings within a data block.

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var b as BinaryStream = BinaryStream.Open(f)
Var s as string = b.Read(b.Length)
Var texts() as string = GetStringsFromDataMBS(s, 3)
```

Break

Notes: Looks over the bytes in the data block to see if there are useful byte sequences which could be text. Returns an array with all text fragments found.

This function is UTF8 aware and will work well for UTF-8 encoded text fragments.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.4pr1](#)

See also:

- 38.1.36 GetStringsFromDataMBS(data as MemoryBlock, MinLength as Integer = 0) as string() 1112
- 38.1.37 GetStringsFromDataMBS(data as ptr, size as Integer, MinLength as Integer = 0) as string() 1113

38.1.39 GetUnicodeMarkersMBS(kind as Integer) as string

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the unicode marker with the given code.

Example:

```
const EncodingUnknown=0
const EncodingUTF8=1
const EncodingUTF16=2 // native
const EncodingUTF16be=3
const EncodingUTF16le=4
const EncodingUTF32=5 // native
const EncodingUTF32be=6
const EncodingUTF32le=7
```

```
msgbox EncodingToHexMBS(GetUnicodeMarkersMBS(EncodingUTF8))
```

Notes: If you concat strings, RB may convert the string to UTF8.

38.1.40 HasPostfixMBS(Text as String, Prefix as String) as boolean

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether text has given postfix.

Example:

```
Var t as string = "Hello World"
Var p as string = "Hello"
Var w as string = "World"
```

```
if HasPrefixMBS(t, p) then
// ok
else
break
MsgBox "failed?"
end if
```

```
if HasPostfixMBS(t, w) then
// ok
else
break
MsgBox "failed?"
end if
```

```
if HasPrefixMBS(t, w) then
break
```

```

MsgBox "failed?"
else
// ok
end if

if HasPostfixMBS(t, p) then
break
MsgBox "failed?"
else
// ok
end if

Break

```

Notes: Returns true if yes, or false if not.
 We may convert text to UTF-8 to compare.
Blog Entries

- [MBS Xojo Plugins, version 18.5pr6](#)

38.1.41 HasPrefixMBS(Text as String, Prefix as String) as boolean

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether text has given prefix.

Example:

```

Var t as string = "Hello World"
Var p as string = "Hello"
Var w as string = "World"

if HasPrefixMBS(t, p) then
// ok
else
break
MsgBox "failed?"
end if

if HasPostfixMBS(t, w) then
// ok
else
break
MsgBox "failed?"
end if

if HasPrefixMBS(t, w) then
break

```

```

MsgBox "failed?"
else
// ok
end if

if HasPostfixMBS(t, p) then
break
MsgBox "failed?"
else
// ok
end if

Break

```

Notes: Returns true if yes, or false if not.
 We may convert text to UTF-8 to compare.
Blog Entries

- [MBS Xojo Plugins, version 18.5pr6](#)

38.1.42 HexstringMBS(input as string, hexlen as Integer, linelen as Integer, linestart as string, lineend as string, spacer as string, filler as string) as string

Plugin Version: 2.6, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a hexadecimal string.
Example:

```

Var s as string
s=HexstringMBS("Hello World",2,10,"<",">" +endofline,"","0")

```

Notes: Fails if input="" or hexlen<1 or linelen<1 or on low memory conditions.
 If filler is "", filler is set to "0".

Memory requirement is around 5 times the memory for input.

or exactly:

$$\text{mem} = 20 + \text{linelen} * \text{hexlen} * (1 + \text{len}(\text{filler})) + 2 * ((\text{linelen} * \text{hexlen} * \text{len}(\text{filler}) + \text{len}(\text{lineend}) + \text{len}(\text{linestart})) * (\text{len}(\text{input}) * 2 / \text{hexlen}))$$

All strings must be in the encoding you want to have.
 Can raise OutOfMemoryException on low memory.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)

- [MBS Xojo / Real Studio Plugins, version 15.2pr4](#)

38.1.43 IsASCIIStringMBS(s as string) as boolean

Plugin Version: 7.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if this string is an ASCII string.

Notes: False if one of the characters of the string has a numeric value of 128 or higher. (->string is not a 7 bit ASCII string)

True if all bytes are in the valid ASCII range.

See also:

- [38.1.44 IsASCIIStringMBS\(s as string, mode as Integer\) as boolean](#)

1117

38.1.44 IsASCIIStringMBS(s as string, mode as Integer) as boolean

Plugin Version: 3.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if this string is an ASCII string.

Notes: Does not work for 16bit unicode strings.

But it works fine with UTF8.

Mode values:

- 0 False if one of the characters of the string has a numeric value of 128 or higher. (->string is not a 7 bit ASCII string)
- 1 False if one of the characters of the string has a numeric value >128 or <32. (->string may not be printable in ASCII, which may require some kind of Base64 encoding to transfer it.)
- 2 False if one of the characters of the string has a numeric value below 32. (->string may contain line breaks or other control characters)

Added Linux support in v5.1.

See also:

- [38.1.43 IsASCIIStringMBS\(s as string\) as boolean](#)

1117

38.1.45 JaroWinklerDistanceMBS(a as string, b as string) as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the string distance.

Example:

```

Var s1 as string = "Hello"
Var s2 as string = "Hallo"
Var s3 as string = "Helo"
Var s4 as string = "Hello"

// 0 for equal, the more it goes to 1.0

Var d1 as Double = levenshteinDistanceMBS(s1,s1)
Var d2 as Double = levenshteinDistanceMBS(s1,s2)
Var d3 as Double = levenshteinDistanceMBS(s1,s3)
Var d4 as Double = levenshteinDistanceMBS(s1,s4)

Var d5 as Double = jaroWinklerDistanceMBS(s1,s1)
Var d6 as Double = jaroWinklerDistanceMBS(s1,s2)
Var d7 as Double = jaroWinklerDistanceMBS(s1,s3)
Var d8 as Double = jaroWinklerDistanceMBS(s1,s4)

break // check values in debugger

```

Notes: Value is normalized, so 0.0 is equal text and 1.0 is totally unequal.

see also

http://en.wikipedia.org/wiki/Jaro-Winkler_distance

Blog Entries

- [MBS SQLite Extension in version 1.4](#)
- [MBS Real Studio Plugins, version 12.4pr10](#)

38.1.46 LevenshteinDistanceMBS(a as string, b as string) as Double

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Calculates the string distance.

Example:

```

Var s1 as string = "Hello"
Var s2 as string = "Hallo"
Var s3 as string = "Helo"
Var s4 as string = "Hello"

// 0 for equal, the more it goes to 1.0

Var d1 as Double = levenshteinDistanceMBS(s1,s1)

```

```

Var d2 as Double = levenshteinDistanceMBS(s1,s2)
Var d3 as Double = levenshteinDistanceMBS(s1,s3)
Var d4 as Double = levenshteinDistanceMBS(s1,s4)

Var d5 as Double = jaroWinklerDistanceMBS(s1,s1)
Var d6 as Double = jaroWinklerDistanceMBS(s1,s2)
Var d7 as Double = jaroWinklerDistanceMBS(s1,s3)
Var d8 as Double = jaroWinklerDistanceMBS(s1,s4)

break // check values in debugger

```

Notes: Value is normalized, so 0.0 is equal text and 1.0 is totally unequal.

see also

http://en.wikipedia.org/wiki/Levenshtein_distance

Blog Entries

- [MBS SQLite Extension in version 1.4](#)
- [MBS Real Studio Plugins, version 12.4pr10](#)

38.1.47 NativeStringMBS(s as string) as string

Plugin Version: 2.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a native string with the given string as content.

Notes: If the string in the parameter s is already native it is returned unchanged and without allocating additional memory. Else the string is copied into a new string with native characters.

On Mac OS Classic native means MacRoman encoding, on Mac OS X it's UTF8 and on Windows the ANSI Codepage.

May return "" on low memory conditions.

Strings and encoding work only perfectly for RB 4.5 or newer.

Added Linux support in v5.1.

38.1.48 RandomBytesStringMBS(Length as Integer, ASCII as boolean=false) as string

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a string with random content.

Example:

```
Var s as string
```

```
s = RandomBytesStringMBS(10, true)
```

MsgBox s

```
s = RandomBytesStringMBS(10, false)
```

MsgBox s

Notes: Length is the number of bytes in the string.
If ASCII is true the string returned in an ASCII string.

Blog Entries

- [CheckUTF8MBS function](#)

Xojo Developer Magazine

- [10.1, pages 71 to 72: Using Plugins, Working with the Monkeybread Plugins by Marc Zeedar](#)

38.1.49 RemoveHTMLTagsMBS(AsciiTextWithTags as string) as string

Plugin Version: 4.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes all html tags from the string.

Example:

```
Var s As String = "<div><P param=""value"">Hello</P></div>"
Var t As String = RemoveHTMLTagsMBS(s)
```

MsgBox t // returns "Hello"

Notes: Returns "" on low memory.
Written to be used with UTF8 strings.
Added Linux support in v5.1.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr2](#)
- [MBS Xojo Plugins, version 19.4pr8](#)

38.1.50 RemoveHTMLTagsWithMBS(AsciiTextWithTags as string, Replacement as string) as string

Plugin Version: 8.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes all html tags from the string and replaces them with the replacement string.

Example:


```
msgbox RemoveHTMLTagsWithMBS("<P>Hello</P>"," ") // returns " Hello "
```

Notes: Returns "" on low memory.
Written to be used with UTF8 strings.

38.1.51 ReplaceLineEndingsMBS(Text as String, NewLine as String, yield as boolean = false) as String

Plugin Version: 19.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces line endings.

Example:

```
Var s as string = "Some text"
Var t as string = ReplaceLineEndingsMBS(s, endofline.Unix)
```

Notes: Whether we got CR, LF, CRLF or even LFCR, we'll replace it with the given replacement text.
If yield is true, we yield to other Xojo threads every 16 MB of input text to keep other threads responsive.

Handles UTF16/UTF32 by converting first to UTF-8.
Other 8 bit encodings (or no encoding set) are passed through.

Returns new text. May raise OutOfMemoryException in case of no memory available.

Blog Entries

- [MBS Xojo Plugins, version 19.0pr7](#)
- [New ReplaceLineEndingsMBS function](#)

38.1.52 ReplaceNonPrintableCharactersMBS(s as string, replacevalue as Integer=46) as string

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces bytes in the string which are not printable with the given byte value.

Example:

```
MsgBox ReplaceNonPrintableCharactersMBS("Hello World")
MsgBox ReplaceNonPrintableCharactersMBS("Hello √$√/° World")
MsgBox ReplaceNonPrintableCharactersMBS("Hello √$√/° World",32)
```

Notes: All bytes in range 32 to 127 are copied and all others replaces with the given byte value. Default is 46 which is a dot.

Returns always an ASCII string.

On low memory this function returns an empty string.

Blog Entries

- [New ReplaceLineEndingsMBS function](#)

38.1.53 ScientificStrMBS(d as Double, digits as Integer) as string

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the given value formatted as scientific number with the given number of digits.

Example:

```
Var d as Double = 3.1415926535897
```

```
MsgBox ScientificStrMBS(d,6)+EndOfLine+ScientificStrMBS(d,9)
```

Blog Entries

- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr5](#)

38.1.54 SQLReplaceBooleanMBS(SQL as string) as string

Plugin Version: 10.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces all false/true strings with 0/1 and handles quotes correctly.

Example:

```
MsgBox SQLReplaceBooleanMBS("INSERT INTO criteria VALUES ('1','9999','0001','000001557',false);")
```

Notes: SQLite used in REALSQLDatabase does not like false and true literals for boolean values. You need to use 0 and 1. So this function helps you converting old queries using false/true. false and true inside a quoted string are not changed.

Blog Entries

- [MBS Plugins 10.3 Release Notes](#)

38.1.55 StrCompBytesMBS(a as string, b as string) as Integer

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares two strings.

Example:

```
msgbox str(StrCompBytesMBS("Hello","hello"))
```

Notes: Returns zero if the two strings are identical.
Returns 1 or -1 if the strings are different.
Empty strings are equal.

Compares the bytes of both strings independent of the text encoding.

So if a="A" in UTF-8 and b="A" in UTF-16 they will not be equal!

38.1.56 StrCompCharactersMBS(a as string, b as string) as Integer

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares two strings.

Example:

```
msgbox str(StrCompCharactersMBS("Hello","hello"))
```

Notes: Returns zero if the two strings are identical.
Returns 1 or -1 if the strings are different.
Empty strings are equal.

Compares the characters of both strings. If the text encodings are not equal, they are both converted to UTF-8 before comparing them.

So if a="A" in UTF-8 and b="A" in UTF-16 they will be equal!

38.1.57 StringANDMBS(a as string,b as string) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies a binary AND on the bytes from the both strings.

Notes: If a is "" or memory is low, the result is "".

If b is shorter as a the b string is used several times.

The function is optimized for several cases, e.g. the case where b is only one, two or four bytes long.

Returns a string with binary encoding. Before using this function, make sure both strings have the same encoding.

Added Linux support in v5.1.

38.1.58 StringIsHTMLreadyMBS(s as string) as boolean

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test whether a string is html safe.

Example:

```
Function html(t as string) As string
#pragma disablebackgroundtasks
#pragma disableautowaitcursor

if StringIsHTMLreadyMBS(t) then
Return t
else
t=EncodingToHTMLMBS(t)
Return ConvertEncoding(t,encodings.UTF8)
end if
End Function
```

Notes: String which are not html safe, need to go through EncodingToHTMLMBS. Using this function saves a lot of time!

Returns true for strings which are html safe.

38.1.59 StringIsXMLreadyMBS(s as string) as boolean

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test whether a string is xml safe.

Notes: String which are not html safe, need to go through EncodingToXMLMBS. Using this function saves a lot of time!

Returns true for strings which are xml safe.

38.1.60 StringORMBS(a as string,b as string) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies a binary OR on the bytes from the both strings.

Notes: If a is "" or memory is low, the result is "".

If b is shorter as a the b string is used several times.

The function is optimized for several cases, e.g. the case where b is only one, two or four bytes long.

Returns a string with binary encoding. Before using this function, make sure both strings have the same

encoding.

Added Linux support in v5.1.

38.1.61 StringXOR2MBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies a binary XOR on the bytes from the both strings.

Example:

```
Var s as string = StringXORMBS("Hello", "World")
Var t as string = StringXOR2MBS("Hello", "World")
```

```
MsgBox EncodeHex(S)+" "+EncodeHex(t)
```

```
Var ss as string = StringXORMBS(s, "World")
Var tt as string = StringXOR2MBS(t, "World")
```

```
MsgBox ss+" "+tt
```

Notes: If data is "" or memory is low, the result is "".

If XorMask is shorter as data the XorMask string is used several times.

Returns a string with binary encoding. Before using this function, make sure both strings have the same encoding.

The difference between StringXORMBS and StringXOR2MBS is that the second version xors also with position of byte in string, so your text looks a little bit more random.

MaskOffset specifies where in the XorMask string to start. This is useful for partial blocks.

Blog Entries

- [MBS Real Studio Plugins, version 12.3pr8](#)

38.1.62 StringXORMBS(data as string, XorMask as string, MaskOffset as Integer = 0) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies a binary XOR on the bytes from the both strings.

Notes: If data is "" or memory is low, the result is "".

If XorMask is shorter as data the XorMask string is used several times.

The function is optimized for several cases, e.g. the case where b is only one, two or four bytes long.

Returns a string with binary encoding. Before using this function, make sure both strings have the same encoding.

MaskOffset specifies where in the XorMask string to start. This is useful for partial blocks.

38.1.63 StrMBS(d as Double) as string

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Formats numbers more human readable

Example:

```
// shows: 0.000000012339 1.234560e-8
MsgBox StrMBS(0.00000001234)+" "+str(0.00000001234)
```

```
// shows 123456789.0 1.234568e+8
MsgBox StrMBS(123456789.0)+" "+str(123456789.0)
```

Notes: The idea is to have a dynamically changing number of digits. We have no scientific notation and up to 15 digits visible (not counting zeros).

Blog Entries

- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr4](#)

38.1.64 UnicodeStringMBS(s as string) as string

Plugin Version: 2.7, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a unicode string with the given string as content.

Notes: If the string in the parameter s is already unicode it is returned unchanged and without allocating additional memory. Else the string is copied into a new string with unicode (16bit) characters.

May return "" on low memory conditions.

Strings and encoding work only perfectly for RB 4.5 or newer.

Added Linux support in v5.1.

The result of this function is unpredictable on bad input strings.
(e.g. no encoding, or encoding does not match the content of the string)

38.1.65 ConvertUnicodeToCharacterCompositionMBS(text as string) as string

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts unicode characters to composed normalization form.

Example:

```
Var s1 as string = ConvertUnicodeToCharacterDecompositionMBS("√§")
```

```
MsgBox s1+": "+EncodeHex(s1)
// shows √§: 61CC88
```

```
Var s2 as string = ConvertUnicodeToCharacterCompositionMBS("√§")
```

```
MsgBox s2+": "+EncodeHex(s2)
// shows √§: C3A4
```

Notes: This function replaces character represented by decomposed representation with the composed representation.

see also

https://en.wikipedia.org/wiki/Unicode_equivalence

Blog Entries

- [MBS Xojo Plugins, version 22.1pr1](#)

38.1.66 ConvertUnicodeToCharacterDecompositionMBS(text as string) as string

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts unicode characters to decomposed normalization form.

Example:

```
Var s1 as string = ConvertUnicodeToCharacterDecompositionMBS("√§")
```

```
MsgBox s1+": "+EncodeHex(s1)
// shows √§: 61CC88
```

```
Var s2 as string = ConvertUnicodeToCharacterCompositionMBS("√§")
```

```
MsgBox s2+": "+EncodeHex(s2)
// shows √§: C3A4
```

Notes: This function replaces character represented by one unicode character by the decomposed variant.

see also

https://en.wikipedia.org/wiki/Unicode_equivalence

Blog Entries

- [MBS Xojo Plugins, version 22.1pr1](#)
- [MBS Xojo Plugins, version 20.1pr4](#)

38.1.67 DecodingFromCP1252MBS(s as string) as string

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with ISO 8859 encoding.

Notes: The source string must be really an CP1252 string!

The string returned is Unicode and you may try NativeString to make a string you can display better.

May return "" on low memory conditions.

Strings and encoding work only perfectly for RB 4.5 or newer.

Added Linux support in v5.1.

38.1.68 DecodingFromHexMBS(s as string) as string

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a Hex string to a binary string.

Example:

`Var s as String`

```
s=EncodingToHexMBS("Hallo")
```

```
MsgBox DecodingFromHexMBS(s)
```

Notes: May return "" on low memory conditions.

Added Linux support in v5.1.

The string returned has no defined string encoding, so use DefineEncoding on the result if needed.

38.1.69 DecodingFromISO8859MBS(s as string) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decodes a string with ISO 8859 encoding.

Example:

`Var s as String`

```
s=EncodingToISO8859MBS("hallo √§√√°")
```

```
MsgBox s
```

```
s=DecodingFromISO8859MBS(s)
```


MsgBox s

s=NativeStringMBS(s)

MsgBox s

Notes: The source string must be really an ISO 8859-1 string!
The string returned is Unicode and you may try NativeString to make a string you can display better.

May return "" on low memory conditions.
Strings and encoding work only perfectly for RB 4.5 or newer.
Added Linux support in v5.1.

38.1.70 EncodingToCP1252MBS(s as string) as string

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string encoded with Codepage 1252.

Notes: The string is converted to Unicode (if it's not already Unicode) and then encoded to a CP 1252 string which is returned with encoding set to binary.
(other encodings are available, but currently not in the plugin)

May return "" on low memory conditions.
Strings and encoding work only perfectly for RB 4.5 or newer.
Added Linux support in v5.1.

38.1.71 EncodingToHexMBS(s as string) as string

Plugin Version: 3.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Encodes a binary string to a hex string.

Example:

```
Var s1 as string = "Hello World"
Var s2 as string = EncodingToHexMBS(s1)
```

```
Var t1 as string = "Umlauts √§√/°"
Var t2 as string = EncodingToHexMBS(t1)
```

```
Var u as new UUIDMBS
Var u1 as string = u.ValueString
Var u2 as string = EncodingToHexMBS(u1)
```

`break` // see variables in debugger

Notes: May return "" on low memory conditions.
Added Linux support in v5.1.

38.1.72 EncodingToISO8859MBS(s as string) as string

Plugin Version: 3.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string encoded with ISO 8859-1.

Example:

`Var s as String`

```
s=EncodingToISO8859MBS("hallo √§√√°")
```

```
MsgBox s
```

```
s=DecodingFromISO8859MBS(s)
```

```
MsgBox s
```

```
s=NativeStringMBS(s)
```

```
MsgBox s
```

Notes: The string is converted to Unicode (if it's not already Unicode) and then encoded to a ISO 8859-1 string which is returned with encoding set to binary.
(other encodings are available, but currently not in the plugin)

May return "" on low memory conditions.
Strings and encoding work only perfectly for RB 4.5 or newer.
Added Linux support in v5.1.

38.2 class TextConverterMBS

38.2.1 class TextConverterMBS

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for converting text from one text encoding to other one.

Example:

```
Var s as string = "Hello √§√√o"
Var c as new TextConverterMBS

c.Input = s
c.InputEncoding = "UTF-8"
c.OutputEncoding = "WINDOWS-1252"

c.Convert

Var output as string = c.Output

Break // see in debugger
```

Notes: Wraps iconv library. Please use LoadIconvLibrary function to load it. Without the library, we use fallback code, which just passes through data.

Encoding list with alternative names:

- ANSI_X3.4-1968,ANSI_X3.4-1986,ASCII,CP367,IBM367,ISO-IR-6,ISO646-US,ISO_646.IRV:1991,US,US-ASCII,CSASCII
- UTF-8
- ISO-10646-UCS-2,UCS-2,CSUNICODE
- UCS-2BE,UNICODE-1-1,UNICODEBIG,CSUNICODE11
- UCS-2LE,UNICODELITTLE
- ISO-10646-UCS-4,UCS-4,CSUCS4
- UCS-4BE
- UCS-4LE
- UTF-16
- UTF-16BE
- UTF-16LE
- UTF-32
- UTF-32BE
- UTF-32LE
- UNICODE-1-1-UTF-7,UTF-7,CSUNICODE11UTF7
- UCS-2-INTERNAL

- UCS-2-SWAPPED
- UCS-4-INTERNAL
- UCS-4-SWAPPED
- C99
- JAVA
- CP819,IBM819,ISO-8859-1,ISO-IR-100,ISO8859-1,ISO_8859-1,ISO_8859-1:1987,L1,LATIN1,CSISOLATIN1
- ISO-8859-2,ISO-IR-101,ISO8859-2,ISO_8859-2,ISO_8859-2:1987,L2,LATIN2,CSISOLATIN2
- ISO-8859-3,ISO-IR-109,ISO8859-3,ISO_8859-3,ISO_8859-3:1988,L3,LATIN3,CSISOLATIN3
- ISO-8859-4,ISO-IR-110,ISO8859-4,ISO_8859-4,ISO_8859-4:1988,L4,LATIN4,CSISOLATIN4
- CYRILLIC,ISO-8859-5,ISO-IR-144,ISO8859-5,ISO_8859-5,ISO_8859-5:1988,CSISOLATINCYRILLIC
- ARABIC,ASMO-708,ECMA-114,ISO-8859-6,ISO-IR-127,ISO8859-6,ISO_8859-6,ISO_8859-6:1987,CSISOLATINARABIC
- ECMA-118,ELOT_928,GREEK,GREEK8,ISO-8859-7,ISO-IR-126,ISO8859-7,ISO_8859-7,ISO_8859-7:1987,ISO_8859-7:2003,CSISOLATINGREEK
- HEBREW,ISO-8859-8,ISO-IR-138,ISO8859-8,ISO_8859-8,ISO_8859-8:1988,CSISOLATINHEBREW
- ISO-8859-9,ISO-IR-148,ISO8859-9,ISO_8859-9,ISO_8859-9:1989,L5,LATIN5,CSISOLATIN5
- ISO-8859-10,ISO-IR-157,ISO8859-10,ISO_8859-10,ISO_8859-10:1992,L6,LATIN6,CSISOLATIN6
- ISO-8859-11,ISO8859-11,ISO_8859-11
- ISO-8859-13,ISO-IR-179,ISO8859-13,ISO_8859-13,L7,LATIN7
- ISO-8859-14,ISO-CELTIC,ISO-IR-199,ISO8859-14,ISO_8859-14,ISO_8859-14:1998,L8,LATIN8
- ISO-8859-15,ISO-IR-203,ISO8859-15,ISO_8859-15,ISO_8859-15:1998,LATIN-9
- ISO-8859-16,ISO-IR-226,ISO8859-16,ISO_8859-16,ISO_8859-16:2001,L10,LATIN10
- KOI8-R,CSKOI8R
- KOI8-U
- KOI8-RU
- CP1250,MS-EE,WINDOWS-1250
- CP1251,MS-CYRL,WINDOWS-1251
- CP1252,MS-ANSI,WINDOWS-1252
- CP1253,MS-GREEK,WINDOWS-1253
- CP1254,MS-TURK,WINDOWS-1254
- CP1255,MS-HEBR,WINDOWS-1255

- CP1256,MS-ARAB,WINDOWS-1256
- CP1257,WINBALTRIM,WINDOWS-1257
- CP1258,WINDOWS-1258
- 850,CP850,IBM850,CSPC850MULTILINGUAL
- 862,CP862,IBM862,CSPC862LATINHEBREW
- 866,CP866,IBM866,CSIBM866
- CP1131
- MAC,MACINTOSH,MACROMAN,CSMACINTOSH
- MACCENTRALEUROPE
- MACICELAND
- MACCROATIAN
- MACROMANIA
- MACCYRILLIC
- MACUKRAINE
- MACGREEK
- MACTURKISH
- MACHEBREW
- MACARABIC
- MACTHAI
- HP-ROMAN8,R8,ROMAN8,CSHPROMAN8
- NEXTSTEP
- ARMSCII-8
- GEORGIAN-ACADEMY
- GEORGIAN-PS
- KOI8-T
- CP154,CYRILLIC-ASIAN,PT154,PTCP154,CSPTCP154
- KZ-1048,RK1048,STRK1048-2002,CSKZ1048
- MULELAO-1
- CP1133,IBM-CP1133

- ISO-IR-166,TIS-620,TIS620,TIS620-0,TIS620.2529-1,TIS620.2533-0,TIS620.2533-1
- CP874,WINDOWS-874
- VISCI,VISCI1.1-1,CSVISCI
- TCVN,TCVN-5712,TCVN5712-1,TCVN5712-1:1993
- ISO-IR-14,ISO646-JP,JIS_C6220-1969-RO,JP,CSISO14JISC6220RO
- JISX0201-1976,JIS_X0201,X0201,CSHALFWIDTHKATAKANA
- ISO-IR-87,JIS0208,JIS_C6226-1983,JIS_X0208,JIS_X0208-1983,JIS_X0208-1990,X0208,CSISO87JISX0208
- ISO-IR-159,JIS_X0212,JIS_X0212-1990,JIS_X0212.1990-0,X0212,CSISO159JISX02121990
- CN,GB_1988-80,ISO-IR-57,ISO646-CN,CSISO57GB1988
- CHINESE,GB_2312-80,ISO-IR-58,CSISO58GB231280
- CN-GB-ISOIR165,ISO-IR-165
- ISO-IR-149,KOREAN,KSC_5601,KS_C_5601-1987,KS_C_5601-1989,CSKSC56011987
- EUC-JP,EUCJP,EXTENDED_UNIX_CODE_PACKED_FORMAT_FOR_JAPANESE,CSEUCPKDFMTJAPANESE
- MS_KANJI,SHIFT-JIS,SHIFT_JIS,SJIS,CSSHIFTJIS
- CP932
- ISO-2022-JP,CSISO2022JP
- ISO-2022-JP-1
- ISO-2022-JP-2,CSISO2022JP2
- CP50221,ISO-2022-JP-MS
- CN-GB,EUC-CN,EUCCN,GB2312,CSGB2312
- GBK
- CP936,MS936,WINDOWS-936
- GB18030
- ISO-2022-CN,CSISO2022CN
- ISO-2022-CN-EXT
- HZ,HZ-GB-2312
- EUC-TW,EUCTW,CSEUCTW
- BIG-5,BIG-FIVE,BIG5,BIGFIVE,CN-BIG5,CSBIG5
- CP950

- BIG5-HKSCS:1999
- BIG5-HKSCS:2001
- BIG5-HKSCS:2004
- BIG5-HKSCS,BIG5-HKSCS:2008,BIG5HKSCS
- EUC-KR,EUCKR,CSEUCKR
- CP949,UHC
- CP1361,JOHAB
- ISO-2022-KR,CSISO2022KR
- CP856
- CP922
- CP943
- CP1046
- CP1124
- CP1129
- CP1161,IBM-1161,IBM1161,CSIBM1161
- CP1162,IBM-1162,IBM1162,CSIBM1162
- CP1163,IBM-1163,IBM1163,CSIBM1163
- DEC-KANJI
- DEC-HANYU
- 437,CP437,IBM437,CSPC8CODEPAGE437
- CP737
- CP775,IBM775,CSPC775BALTIC
- 852,CP852,IBM852,CSPC852
- CP853
- 855,CP855,IBM855,CSIBM855
- 857,CP857,IBM857,CSIBM857
- CP858
- 860,CP860,IBM860,CSIBM860
- 861,CP-IS,CP861,IBM861,CSIBM861

- 863,CP863,IBM863,CSIBM863
- CP864,IBM864,CSIBM864
- 865,CP865,IBM865,CSIBM865
- 869,CP-GR,CP869,IBM869,CSIBM869
- CP1125
- EUC-JIS-2004,EUC-JISX0213
- SHIFT_JIS-2004,SHIFT_JISX0213
- ISO-2022-JP-2004,ISO-2022-JP-3
- BIG5-2003
- ISO-IR-230,TDS565
- ATARI,ATARIST
- RISCOS-LATIN1

Blog Entries

- [MBS Xojo Plugins, version 23.6pr2](#)
- [News from the MBS Xojo Plugins Version 20.4](#)
- [MBS Xojo Plugins, version 20.4pr9](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 19.4](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 19.1](#)
- [MBS Xojo Plugins, version 19.1pr3](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [17.5, page 37: What's New in the MBS Plugins, With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes](#)
- [17.3, page 11: News](#)

38.2.2 Methods

38.2.3 Canonicalize(name as String) as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Canonicalize an encoding name.

Example:

```
MsgBox TextConverterMBS.Canonicalize("utf-8")
MsgBox TextConverterMBS.Canonicalize("Latin1")
```

Notes: The result is either a canonical encoding name, or name itself.

38.2.4 Constructor

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.

Notes: Currently enables translit and ignore by default.

38.2.5 Convert

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts text.

Example:

```
Var s as string = "Hello √§√√°"
Var c as new TextConverterMBS

c.Input = s
c.InputEncoding = "UTF-8"
c.OutputEncoding = "WINDOWS-1252"

c.Convert

Var output as string = c.Output

Break // see in debugger
```

Notes: Input is read from either Input or InputData properties.

ErrorCode and ErrorMessage are set.

Output is set with output data.

38.2.6 EncodingNames(Mode as Integer = 0) as String()

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries names of text encodings.

Example:

```
MsgBox Join(TextConverterMBS.EncodingNames(1),EndOfLine)
```

Notes: Mode 0 returns list of text encoding names.

Mode 1 returns list of text encoding names and alternative names separated by comma.

Mode 2 returns list of text encoding names and alternative names.

e.g. "CP1252,MS-ANSI,WINDOWS-1252" in the list for mode 1.

38.2.7 LoadIconvLibrary(path as String, byref Error as String) as boolean

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads the iconv library.

Notes: The TextConverterMBS class uses libiconv for text encoding conversion.

If you explicitly need, you can load the library on start of solution.

MBS Plugin may try to load iconv.dll/dylib/so automatically when first iconv function is called.

38.2.8 Properties

38.2.9 ErrorCode as Integer

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Last error code from Convert method.

Notes: (Read and Write property)

38.2.10 ErrorMessage as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Text message for error.

Notes: (Read and Write property)

38.2.11 Ignore as Boolean

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to ignore errors.

Notes: Characters that cannot be represented in the target character set will be silently discarded.

(Read and Write property)

38.2.12 Input as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Input text string.

Notes: (Read and Write property)

38.2.13 InputData as MemoryBlock

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Input as memoryblock.

Notes: (Read and Write property)

38.2.14 InputEncoding as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Name of input encoding.

Notes: e.g. "UTF-8"

(Read and Write property)

38.2.15 InputPosition as Integer

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The input position on the end of convert.

Notes: If errors are not ignored, this shows error position.

(Read and Write property)

38.2.16 LibVersion as Integer

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The library version for iconv.

Notes: Returns major * 256 + minor, e.g. &h10B for 1.11.

(Read only property)

38.2.17 Output as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The output text.

Notes: String may not have an encoding set, if the plugin does not recognize the encoding.

But for e.g. UTF-8, we set the text encoding.

(Read and Write property)

38.2.18 OutputEncoding as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Name of output encoding.

Notes: e.g. "UTF-8"

(Read and Write property)

38.2.19 OutputLength as Integer

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The length of the output in bytes.

Notes: (Read and Write property)

38.2.20 Translit as Boolean

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to activate transliteration.

Notes: This means that when a character cannot be represented in the target character set, it can be

approximated through one or several characters that look similar to the original character.
(Read and Write property)

38.3 class TextEncoding

38.3.1 class TextEncoding

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Built in class from Xojo for text encodings.

38.3.2 Methods

38.3.3 InternetNameMBS as string

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the name of this text encoding.

Notes: We have this method in MBS Plugin as InternetName function does not work on Linux/Windows. (See feedback case 30712)

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 13.5pr6](#)

Chapter 39

System

39.1 Globals

39.1.1 CrashNiceMBS

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Crashes the application.

Notes: Does a nil object dereference which leads to a crash.

39.1.2 CrashUglyMBS

Plugin Version: 4.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Crashes the application.

Notes: This function should hide the stack list so it should be impossible to know what function crashed.

It may take a second to finish it, but CrashReporter (on Mac OS X) and other tools will not be able to show you which function crashed.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.2pr1](#)

39.1.3 DelayMBS(time as Double)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Stops the application for the given time.

Example:

DelayMBS 1.5 // wait one and a half seconds

Notes: The application should take nearly 0% CPU Power till time has gone.
If you need more options like blocking other RB threads, check the DelayMBS method with mode parameter.

Another example from Kirk Clendinning:

Property: lastupdate as Integer

```
Sub YieldProcessorTime(periodticks as Integer, widthmilliseconds as Integer)
// use the Monkeybread Software delay to yeild time for a screen update
// every period*ticks for a pulse width of width milliseconds
```

```
if ticks>lastupdate+periodTicks then
delayMBS widthmilliseconds/1000
lastupdate=ticks
end if
```

```
exception err
ExceptionReport(err, "Utils:Methods:YieldProcessorTime")
end sub
```

YieldProcessorTime(15, 20) gives me a delay of 20ms every 250ms. But, since the delayMBS yields time more efficiently than using a RB thread, this turns out to be a much more effective way to update progress bars etc.
I don't need the delay per se, but since RB still hasn't written a good yield function (that I know of) your delayMSB works instead.

Please use optional mode flags with DelayDontWaitNextEvent when using on thread.

Using DelayMBS on Web Applications can cause problems as the server is paused than for all clients, a really bad thing.

See also:

- 39.1.4 DelayMBS(time as Double, mode as Integer)

1144

39.1.4 DelayMBS(time as Double, mode as Integer)

Plugin Version: 6.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Stops the application for the given time.

Example:

DelayMBS 1.5, 0 // wait one and a half seconds

Notes: Like DelayMBS with one parameter but additional mode parameter:

combine the following constants:

```
const DelayDontMPYield = 1      // Don't give additional time for other OS threads
const DelayDontRBYield = 2     // Don't give Xojo time for other Xojo threads
const DelayDontThreadYield = 4 // Don't give addition time for other OS threads
const DelayDontQuickTimeYield = 8 // Don't give time for QuickTime Movieplayer
const DelayDontWait = 16       // Don't give CPU time for other OS threads
const DelayDontWaitNextEvent = 32 // Don't give CPU time to handle events on Mac OS X to avoid beachball
                                cursor
const DelayDontSleep = 64      // Don't give CPU time for other OS threads (Windows)
```

Using DelayDontWaitNextEvent may show the beach ball cursor, but will have the OpenDocument event working as other AppleEvents. Please use DelayDontWaitNextEvent option for running on threads.

Using DelayMBS on Web Applications can cause problems as the server is paused than for all clients, a really bad thing.

Blog Entries

- [MBS Xojo Plugins, version 22.1pr1](#)
- [MBS Xojo / Real Studio Plugins, version 16.0pr8](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr5](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr4](#)
- [MBS Real Studio Plugins, version 13.1pr7](#)

Xojo Developer Magazine

- [19.2, page 33: Shake It!, Shake any window to show an incorrect entry... on any platform, without plugins or declares! by Markus Winter](#)

See also:

- [39.1.3 DelayMBS\(time as Double\)](#)

1143

39.1.5 SleepMBS(time as Double)

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sleeps the current thread for the given time in seconds.

Example:

```
Var sock as TCPSocket // some socket
Var done as Boolean // some global property
```

```
do
```

```
SleepMBS(0.1) // 100 ms
sock.poll
```

```
loop until done
```

Notes: Sleep does not yield CPU time to other functions, so you really sleep the whole process. Useful for console projects which sleep for a few milliseconds in a loop polling a socket.

If you need to give CPU time to socket, other threads or for quicktime movie playback, please use DelayMBS.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr9](#)

39.1.6 AbortMBS

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The abort function causes abnormal program termination to occur.

Notes: Any open streams are flushed and closed.

The abort function never returns.

39.1.7 ExitMBS(code as Integer)

Plugin Version: 8.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Perform normal program termination.

Notes: See "man 3 exit" on the terminal on Mac OS X or Linux for details.

You define your own error codes. Zero is no error by convention.

Blog Entries

- [Console and GUI in one project](#)
- [Tipp of the day: Shell SSH Trouble and Workaround](#)

39.1.8 BacktraceMBS(MaxFrames as Integer = 0, skip as Integer = 2) as string()

Plugin Version: 13.0, Platforms: macOS, Linux, Targets: All.

Function: Queries the stack trace of the current function.

Example:

```
MsgBox Join(BacktraceMBS, EndOfLine)
```

Notes: MaxFrames: How many steps to show at maximum. Default is 128 currently.

skip: How many entries to skip. Default is 2 to not show this function.

Requires Mac OS X 10.5 or Linux.

Skip is 2 to hide this plugin function in the list.

Blog Entries

- [MBS Xojo Plugins, version 20.1pr1](#)
- [Updater Kit 1.5 and BugReporter Kit 1.1](#)
- [MBS Real Studio Plugins, version 13.0pr1](#)

39.1.9 MillisecondsMBS as Double

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries current milliseconds time.

Notes: Like Microseconds in Xojo, but queries system functions for milliseconds counter.

39.1.10 GetDoubleClickIntervalMBS as Integer

Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the time in ticks which makes two clicks to one double click.

Example:

```
sub MouseUp(X as Integer, Y as Integer)
  Var currentClickTicks as Integer
  currentClickTicks = ticks

  if (currentClickTicks - lastClickTicks) <= GetDoubleClickIntervalMBS then
  if abs(x - lastClickX) <= 5 and abs(y - lastClickY) <= 5 then
    DoubleClick //Fire new Doubleclick event
  return
end
end
```

```
lastClickTicks = currentClickTicks
lastClickX = x
lastClickY = y
MouseUp
End sub
```

Notes: Time returned in ticks. One tick is 1/60 of a second.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 14.0pr1](#)

39.2 class SignalHandlerMBS

39.2.1 class SignalHandlerMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: A signal handler for an unix signal.

Notes: You can use this class to catch signals like SIGALRM oder SIGQUIT.

See

[http://en.wikipedia.org/wiki/Signal_\(computing\)](http://en.wikipedia.org/wiki/Signal_(computing))

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 25.0](#)
- [MBS Xojo Plugins, version 24.6pr3](#)
- [MBS Xojo Plugins, version 24.1pr5](#)
- [Xojo Web signal handling](#)
- [MBS Xojo / Real Studio Plugins, version 14.2pr5](#)
- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.0](#)
- [MBS Xojo / Real Studio Plugins, version 14.0pr3](#)
- [MBS REALbasic Plugins, version 10.6pr11](#)

Xojo Developer Magazine

- [12.2, page 10: News](#)

39.2.2 Methods

39.2.3 alarm(seconds as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: Set signal timer alarm.

Example:

```
SignalHandlerMBS.alarm 1
```

Notes: The alarm() function sets a timer to deliver the signal SIGALRM to the calling process after the specified number of seconds. If an alarm has already been set with alarm() but has not been delivered, another call to alarm() will supersede the prior call. The request alarm(0) voids the current alarm and the signal SIGALRM will not be delivered.

Due to setitimer restriction the maximum number of seconds allowed is 100000000.

alarm sends signal 14 after the time out.

39.2.4 ClearFlag(signalIndex as Integer)

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears the flag with given index.

Notes: Please call this after you got IsFlagSet.

39.2.5 ClearFlags

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears all flags.

39.2.6 ClearStacktrace(signalIndex as Integer)

Plugin Version: 14.2, Platforms: macOS, Linux, Targets: All.

Function: Clears stored stack trace for the signal with given index.

39.2.7 Close

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: Releases this object's resources.

39.2.8 IsFlagSet(signalIndex as Integer) as boolean

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if a given flag was set.

39.2.9 QueryStacktrace(signalIndex as Integer, skip as Integer = 2) as string()

Plugin Version: 14.2, Platforms: macOS, Linux, Targets: All.

Function: Queries stack at the time the signal handler was triggered.

Notes: Useful to know later which code got the signal.

Works for flag handlers and for the event handlers if enabled.

39.2.10 SendSignal(PID as Integer, Signal as Integer) as boolean

Plugin Version: 11.0, Platforms: macOS, Linux, Targets: All.

Function: Sends a signal.

Example:

```
// send signal to a process
// PID is 17779 here
// 9 means kill on Mac OS X
call SignalHandlerMBS.SendSignal(17779, 9)
```

Notes: Returns true on success and false on failure.

Without a signal handler the target app could quit.

Lookup signal.h for the list of signals. They can be different on Mac and Linux.

39.2.11 SendSignalToSelf(Signal as Integer) as boolean

Plugin Version: 11.0, Platforms: macOS, Linux, Targets: All.

Function: Send a signal to the current process

Notes: Returns true on success and false on failure.
Without a signal handler the target app could quit.

39.2.12 SetDefaultHandler(signalIndex as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: Defines to use the default handler for the given signal number.

Example:

```
Const SIGALRM = 14
if SignalHandlerMBS.SetDefaultHandler(SIGALRM) then
  MsgBox "Done"
else
  MsgBox "Failed"
end if
```

Notes: Returns true on success and false on failure.

39.2.13 SetEventHandler(signalIndex as Integer, CollectStackTrace as boolean = false) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: Defines to call the Signal Event whenever a signal with the given number is received.

Notes: Returns true on success and false on failure.

If CollectStackTrace is true, you can later use QueryStacktrace to get the stack trace from the time the signal was captured.

39.2.14 SetFlagHandler(signalIndex as Integer, CollectStackTrace as boolean = false) as boolean

Plugin Version: 14.0, Platforms: macOS, Linux, Targets: All.

Function: Defines to set the flag whenever a signal with the given number is received.

Notes: Returns true on success and false on failure.

Please use IsFlagSet in your app regularly to check if the flag was set.

If CollectStackTrace is true, you can later use QueryStacktrace to get the stack trace from the time the signal was captured.

39.2.15 SetIgnore(signalIndex as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: Defines to ignore the event with the given number.

Example:

```
Const SIGALRM = 14
if SignalHandlerMBS.SetIgnore(SIGALRM) then
MsgBox "Done"
else
MsgBox "Failed"
end if
```

Notes: Returns true on success and false on failure.

39.2.16 SetPrintBacktraceAndAbortHandler(signalIndex as integer) as boolean

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines to use the plugin handler for the given signal number.

Example:

```
Call SignalHandlerMBS.SetPrintBacktraceAndAbortHandler(4) // illegal instruction
Call SignalHandlerMBS.SetPrintBacktraceAndAbortHandler(10) // bus error
Call SignalHandlerMBS.SetPrintBacktraceAndAbortHandler(11) // segmentation fault

// now cause a crash
Var p As ptr = Nil
p.Int32(0) = 0
```

Notes: The signal is handled by the plugin. We print the backtrace and abort. We write details to a crash.txt file in the current folder.

39.2.17 SignalStatus(signalIndex as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: Queries the signal state.

Notes: Returns 0 (unknown), kSignalDefault, kSignalEvent, kSignalFlag or kSignalIgnored.

39.2.18 Events

39.2.19 Signal(n as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Targets: All.

Function: A signal was called and you set an event handler.

Example:

```

Sub Signal(n as Integer)
// restore system handler in case we get that crash again!

MySignalHandlerMBS.SetDefaultHandler n

// Mac and Linux can have different signal numbers:

#if TargetMacOS then
Var c as string = "Signal "+str(n)+" on Mac OS X"
#elseif TargetLinux then
Var c as string = "Signal "+str(n)+" on Linux"
#else
Var c as string = "Signal "+str(n)+" on ?"
#endif

Var BackTraceLines() as string
#if mbs.BuildNumber>17662 and not TargetWin32 then // new in 13.0 plugins
BackTraceLines = BacktraceMBS
#endif

// show your bug reporter (or the MBS one)
'dim b as new Bugreporter
'b.ShowExceptionReporter c, BackTraceLines

// quit now without cleaning up the RB runtime which may crash again
ExitMBS 1
End Sub

```

Notes: This function may be called at any time, so you need to use code which is thread safe. (see ThreadMBS class)

39.2.20 Constants

Signal Handler Modes

Constant	Value	Description
kSignalDefault	1	The signal calls default handler.
kSignalEvent	3	The signal raises an event and sets the flag.
kSignalFlag	4	The signal sets a flag.
kSignalIgnored	2	The signal is ignored.
kSignalPrintBacktraceAndAbort	8	The signal is handled by the plugin. We print the backtrace and abort.

39.3 module SystemInformationMBS

39.3.1 module SystemInformationMBS

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: A module to collect all the system information functions.

Example:

```
msgbox "Welcome "+SystemInformationMBS.Username+"!"
```

Blog Entries

- [News from the MBS Xojo Plugins Version 24.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.3](#)
- [More Menubar options in macOS Sonoma](#)
- [News from the MBS Xojo Plugins Version 23.3](#)
- [News from the MBS Xojo Plugins Version 21.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.3](#)
- [News from the MBS Xojo Plugins Version 20.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.3](#)
- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4](#)
- [New for Mac OS X 10.10 in MBS Xojo Plugins](#)

Xojo Developer Magazine

- [7.4, page 8: News](#)
- [20.5, page 10: News](#)
- [19.5, page 9: News](#)
- [13.6, page 41: A Multicore Processing Primer, Part One by Markus Winter](#)

39.3.2 Methods

39.3.3 AvailableRAM as Double

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the size of the available memory.

Example:

```
msgBox format(SystemInformationMBS.AvailableRAM/1024/1024,"0")+ " MB of RAM free."
```

Notes: This function is useful if you want to know how much memory you can allocate without the system swapping.

On Windows, Mac OS X and Linux you can normally allocate up to 2 or 3 GB of memory in your address-room. A swap file on disc is used to cache memory which does not fit into physical memory.

So if you need to allocate a buffer to store temporary data, you can use this function to calculate a good size.

For example a file copy function could use something like this:

```
// a quarter of free memory is good to leave room for some OS buffers:
buffersize=SystemInformationMBS.AvailableRAM/4
```

```
// minimum 4 MB
if buffersize<1024*1024*4 then
buffersize<1024*1024*4
end if
```

```
// maximum 128 MB to make chunks not too big and application too irresponsible
if buffersize<1024*1024*128 then
buffersize<1024*1024*128
end if
```

In older plugins this function was named AvailableRAMMBS.

39.3.4 BusSpeed as Double

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the speed of the System bus in Hz.

Example:

```
msgbox "Your main CPU has "+format(SystemInformationMBS.BusSpeed/1000000.0,"0")+ " MHz"
```

39.3.5 Computername as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the name of the computer.

Example:

```
msgbox "Hi, I'm "+SystemInformationMBS.Computername+""
```

Notes: On Mac OS, the name is queried in this order:

1. Try to ask the Mac OS X Corefoundation for the name.
2. Try to ask via AppleEvents the Finder or FileSharing.
3. Read it from the system resources.

On Linux or Windows the system name.

In older plugins this function was named ComputerNameMBS.

39.3.6 CPUBrandString as string

Plugin Version: 10.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the CPU brand string.

Example:

```
MsgBox SystemInformationMBS.CPUBrandString
```

Notes: If it is an Intel x86 Chip or something compatible, you get strings like this: "Intel(R) Core(TM) i7 CPU M 620 @ 2.67GHz" (Same as CPUIDMBS.BrandString). But on PowerPC chips we return strings like "PowerPC G5 (970MP)" (same as CPUInfoMBS.CPUName).

May return "" if the processor is unknown.

39.3.7 CPUSpeed as Double

Plugin Version: 11.1, Platform: macOS, Targets: All.

Function: Returns the Speed of the main CPU in Hz.

Example:

```
msgbox "Your main CPU has "+format(SystemInformationMBS.CPUSpeed/1000000.0,"0")+ " MHz"
```

39.3.8 DomainName as string

Plugin Version: 10.1, Platform: Windows, Targets: All.

Function: Returns the domain name of a Windows PC.

Example:

```
msgbox SystemInformationMBS.DomainName
```

Notes: Returns an empty string on any error.

39.3.9 HardDiscSerial as string

Plugin Version: 10.3, Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the hard disc serial number of the first hard disc which has a serial number.

Example:

```
MsgBox SystemInformationMBS.HardDiscSerial
```

Notes: This function can return "" if nothing is found.

Seems to return always empty string on Mac OS X 10.4 as the system properties dictionary does not contain the serial number there.

On Windows this function sometimes returns empty string, but later works again on the same machine. Reason unknown.

In Windows 8.1 it looks like `WindowsWMIMBS.InitSecurity(false)` must be called at app.open time as Xojo will do some things in background when opening first window which block our queries.

39.3.10 HostName as string

Plugin Version: 10.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the hostname for this computer.

Example:

```
MsgBox SystemInformationMBS.HostName
```

39.3.11 Is64bitWindows as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the Windows you are using is a 64-bit Windows.

Example:

```
msgbox str(SystemInformationMBS.Is64bitWindows)
```

Notes: Returns true for x64 Windows editions for 32-bit applications.
This function could also be named isWoW64 for "is Windows on Windows 64 bit."
Only for Intel CPU.
Will return false if used on ARM CPU.

Returns always true on 64bit target.

39.3.12 isARM as Boolean

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether CPU is ARM.

Example:

```
If TargetMacOS Then
If SystemInformationMBS.isARM Then
MsgBox "Running ARM version on ARM CPU."
Elseif SystemInformationMBS.IsTranslated Then
MsgBox "Running Intel version on ARM CPU."
Else
MsgBox "Running Intel version on Intel CPU."
End If
End If
```

Notes: Returns true for:

- Raspberry Pi for Linux ARM
- iOS or Android on device.
- Apple Silicon Mac

Returns false for x86 CPUs, e.g. MacOS as well as iOS/Android in simulator.

39.3.13 isBigSur(orHigher as boolean = true) as boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is macOS 11 Big Sur or newer.

Notes: Returns 1 if this is Big Sur or newer, otherwise 0.

39.3.14 isCatalina(orHigher as boolean = true) as boolean

Plugin Version: 19.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is MacOS 10.15 Catalina or newer.

Example:

```
MsgBox str(SystemInformationMBS.isMojave)
```

Notes: Returns true on MacOS 10.15 or newer.

39.3.15 isElCapitan(orHigher as boolean = true) as boolean

Plugin Version: 15.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.11 El Capitan or newer.

Example:

```
MsgBox str(SystemInformationMBS.isElCapitan)
```

Notes: Returns true on Mac OS X 10.11.

If orHigher is set, it will also return true on 10.12.

39.3.16 isHighSierra(orHigher as boolean = true) as boolean

Plugin Version: 18.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is macOS 10.13 High Sierra or newer.

Example:

```
MsgBox str(SystemInformationMBS.isHighSierra)
```

Notes: Returns true on Mac OS X 10.13.

If orHigher is set, it will also return true on 10.14.

39.3.17 isLeopard(orHigher as boolean = true) as boolean

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.5 Leopard or newer.

Example:


```
MsgBox str(SystemInformationMBS.isLeopard)
```

Notes: Returns true on Mac OS X 10.5 or newer.

39.3.18 isLion(orHigher as boolean = true) as boolean

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.7 Lion or newer.

Example:

```
MsgBox str(SystemInformationMBS.isLion)
```

Notes: Returns true on Mac OS X 10.7 or newer.

39.3.19 isMacOSX as Boolean

Plugin Version: 8.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if being called on Mac OS X.

Example:

```
msgbox str(SystemInformationMBS.isMacOSX)
```

39.3.20 isMavericks(orHigher as boolean = true) as boolean

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.9 Mavericks or newer.

Example:

```
MsgBox str(SystemInformationMBS.isMavericks)
```

Notes: Returns true on Mac OS X 10.9 or newer.

39.3.21 isMojave(orHigher as boolean = true) as boolean

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is MacOS 10.14 Mojave or newer.

Example:

```
MsgBox str(SystemInformationMBS.isMojave)
```

Notes: Returns true on MacOS 10.14 or newer.

39.3.22 isMonterey(orHigher as boolean = true) as boolean

Plugin Version: 21.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is macOS 12 Monterey or newer.

Notes: Returns 1 if this is Monterey or newer, otherwise 0.

39.3.23 isMountainLion(orHigher as boolean = true) as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.8 Mountain Lion or newer.

Example:

```
MsgBox str(SystemInformationMBS.isMountainLion)
```

Notes: Returns true on Mac OS X 10.8 or newer.

39.3.24 isSequoia(orHigher as boolean = true) as boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is macOS 15 Sequoia or newer.

Example:

```
If SystemInformationMBS.isSequoia Then  
  MsgBox "Is Sequoia or heigher"  
Else  
  MsgBox "Is older version"  
End If
```

Notes: Returns true if this is Sequoia or newer, otherwise false.

39.3.25 isSierra(orHigher as boolean = true) as boolean

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is macOS 10.12 Sierra or newer.

Example:

```
MsgBox str(SystemInformationMBS.isSierra)
```

Notes: Returns true on Mac OS X 10.12.
If orHigher is set, it will also return true on 10.13.

39.3.26 isSnowLeopard(orHigher as boolean = true) as boolean

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.6 Snow Leopard or newer.

Example:

```
MsgBox str(SystemInformationMBS.isSnowLeopard)
```

Notes: Returns true on Mac OS X 10.6 or newer.

39.3.27 isSonoma(orHigher as boolean = true) as boolean

Plugin Version: 23.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is macOS 14 Sonoma or newer.

Example:

```
If SystemInformationMBS.isSonoma Then  
  MsgBox "Is Sonoma or heigher"  
Else  
  MsgBox "Is older version"  
End If
```

Notes: Returns true if this is Sonoma or newer, otherwise false.

39.3.28 IsTranslated as Integer

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries whether application is translated on Apple Silicon Macs.

Example:

```
If TargetMacOS Then
If SystemInformationMBS.isARM Then
MsgBox "Running ARM version on ARM CPU."
ElseIf SystemInformationMBS.IsTranslated Then
MsgBox "Running Intel version on ARM CPU."
Else
MsgBox "Running Intel version on Intel CPU."
End If
End If
```

Notes: Returns 1 if Intel code gets translated to ARM code.

Returns 0 for a native ARM application.

Returns -1 if unknown.

39.3.29 isVentura(orHigher as boolean = true) as boolean

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Whether the operation system is macOS 13 Ventura or newer.

Example:

```
If SystemInformationMBS.isVentura(False) Then
MessageBox "Is Ventura exactly."
ElseIf SystemInformationMBS.isVentura(True) Then
MessageBox "Is Ventura or heigher"
Else
MessageBox "Is older version"
End If
```

Notes: Returns true if this is Ventura or newer, otherwise false.

39.3.30 isWindows10(orHigher as boolean = false) as Boolean

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns true if called on Windows 10.

Example:

```
msgbox str(SystemInformationMBS.isWindows10)
```

Notes: And False on Mac OS, Linux, Windows 7/Vista/XP/2000/ME/98/95/8/8.1.
If orHigher, than it returns true if OS Version is newer.

39.3.31 isWindows11(orHigher as boolean = false) as Boolean

Plugin Version: 21.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns true if called on Windows 11.

Example:

```
msgbox str(SystemInformationMBS.isWindows11)
```

Notes: And False on Mac OS, Linux, Windows 7/Vista/XP/2000/ME/98/95/8/8.1/10.
If orHigher, than it returns true if OS Version is newer.

39.3.32 isWindows2000(orHigher as boolean = false) as Boolean

Plugin Version: 8.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns true if called on Windows 2000.

Example:

```
msgbox str(SystemInformationMBS.isWindows2000)
```

Notes: And False on Mac OS X, Linux, Windows 7/XP/Vista/ME/98/95.
If orHigher, than it returns true if OS Version is newer.

39.3.33 isWindows7(orHigher as boolean = false) as Boolean

Plugin Version: 10.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns true if called on Windows 7.

Example:

```
msgbox str(SystemInformationMBS.isWindows7)
```

Notes: And False on Mac OS X, Linux, Windows Vista/XP/2000/ME/98/95.
If orHigher, than it returns true if OS Version is newer.

39.3.34 isWindows8(orHigher as boolean = false) as Boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns true if called on Windows 8.

Example:

```
msgbox str(SystemInformationMBS.isWindows8)
```

Notes: And False on Mac OS X, Linux, Windows 7/Vista/XP/2000/ME/98/95.
If orHigher, than it returns true if OS Version is newer.

39.3.35 isWindows81(orHigher as boolean = false) as Boolean

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns true if called on Windows 8.1

Example:

```
msgbox str(SystemInformationMBS.isWindows81)
```

Notes: And False on Mac OS X, Linux, Windows 7/Vista/XP/2000/ME/98/95/8/10.
If orHigher, than it returns true if OS Version is newer.

39.3.36 isWindowsVista(orHigher as boolean = false) as Boolean

Plugin Version: 8.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns true if called on Windows Vista.

Example:

```
msgbox str(SystemInformationMBS.isWindowsVista)
```

Notes: And False on Mac OS X, Linux, Windows 7/XP/2000/ME/98/95.
If orHigher, than it returns true if OS Version is newer.

39.3.37 isWindowsXP(orHigher as boolean = false) as Boolean

Plugin Version: 8.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns true if called on Windows XP.

Example:

```
msgBox str(SystemInformationMBS.isWindowsXP)
```

Notes: And False on Mac OS X, Linux, Windows 7/Vista/2000/ME/98/95.
If orHigher, than it returns true if OS Version is newer.

39.3.38 isYosemite(orHigher as boolean = true) as boolean

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether the operation system is Mac OS X 10.10 Yosemite or newer.

Example:

```
MsgBox str(SystemInformationMBS.isYosemite)
```

Notes: Returns true on Mac OS X 10.10 or newer.

39.3.39 LogicalRAM as Double

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns the size of the logical installed memory.

Example:

```
msgBox format(SystemInformationMBS.LogicalRAM/1024/1024,"0")+ " MB of RAM built in."
```

Notes: On Windows the total virtual memory size.
On Linux always 4 GB.

In older plugins this function was named LogicalRAMMBS.

39.3.40 MACAddress as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the hardware ethernet address of the first ethernet card.

Example:

```
msgBox SystemInformationMBS.MACAddress
```

Notes: It returns a string with 6 bytes.

Works now with MBS Plugin 2.6 for Mac OS Classic, Carbon inside Classic and Mac OS X, but may not work for a Classic application running on Mac OS X. And may fail on some machines if no OpenTransport is running.

On Windows and Mac OS X you can have multiple ethernet cards and you should use a command line tool with the shell class to find what you need.

For example on Windows:

```
ipconfig /all
```

or on Mac OS X:

```
ifconfig -a
```

You can replace this function with usage of the `NetworkInterface` class in newer RB versions.

Added Linux support in version 16.4.

39.3.41 `MACAddressString` as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the hardware ethernet address of the first ethernet card.

Example:

```
msgBox SystemInformationMBS.MACAddressString
```

Notes: Same as `MACAddress`, but with different format of output.

You can replace this function with usage of the `NetworkInterface` class in newer RB versions.

The plugin asks on Mac OS X the IOKit framework for the primary ethernet interface. We are not sure what Apple really defines for being the primary one, so let's test it:

- Ethernet on and Airport off ->Ethernet MAC Address

- Ethernet off and Airport on ->Ethernet MAC Address

Added Linux support in version 16.4.

39.3.42 MacBoardID as string

Plugin Version: 19.4, Platform: macOS, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Queries board-id for Mac.

Example:

MsgBox SystemInformationMBS.MacBoardID

Notes: e.g. "Mac-1234567890123456"

39.3.43 MacBugFixVersion as Integer

Plugin Version: 10.3, Platform: macOS, Targets: All.

Function: The BugFix part of the Mac OS version number.

Example:

MsgBox str(SystemInformationMBS.MacBugFixVersion)

Notes: The bug fix system version number; in 10.4.17 this would be the decimal value 17.

Intel apps running in Rosetta on Apple M1 will report 10.16 instead of 11.x.

39.3.44 MacHasHardwareAcceleratedCoreImage as boolean

Plugin Version: 8.0, Platform: macOS, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Queries whether hardware acceleration is used for CoreImage.

Example:

```
if SystemInformationMBS.MacHasHardwareAcceleratedCoreImage then
msgbox "CoreImage should be very fast."
else
msgbox "CoreImage may be slow."
end if
```

Notes: Queries OpenGL whether programmable fragments are supported.

39.3.45 MachineID(flags as Integer = 15) as string

Plugin Version: 10.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a Machine ID for the current computer.

Example:

```
// this is how we build it.
Var t as string = "MachineID"+SystemInformationMBS.HardDiscSerial+SystemInformationMBS.MacSerialNumber+SystemInformationMBS.MacModel+SystemInformationMBS.CPUBrandString

// you could extend it with SystemInformationMBS.WinProductKey

Var m as string = MD5StringMBS(t)

MsgBox "Machine ID: "+SystemInformationMBS.MachineID+EndOfLine+"My Machine ID: "+m
```

Notes: Returns a 32 byte long hex string with a Machine ID.

Example value: "EE537483656B25996B51B7F4C99F9083".

This ID is based on the results of the MacSerialNumber, MacModel, CPUBrandString and HardDiscSerial functions. If all 4 functions have no value, the result is always "A2254DEF74A74608D76D1BA49BD2E82A". Also the result could change in future if we fix a bug in one of the functions so that the result values differ.

It is not based on the MACAddressString function as your MAC Address can change when switching between wired and wireless networks. Also we do not check the PhysicalRAM as RAM is a typical thing which changes over time.

You can store this value in some preferences/license file and later compare it against the current value to see if the machine may have changed. In that case ask user to revalidate license, for example by asking for the serial number.

It can happen that 2 PCs have the same MachineID, typical two virtual machines. So this ID is not unique. But it is very likely that two different computers produce different MachineIDs.

Added flags parameter in 14.1:

The Machine ID may be different if one of the components returns a different result. Result may be different in whether app runs as admin or not on Windows.

flagHardDiscSerial	1	Use hard disk serial.
flagMacSerialNumber	2	Use Mac Serial number (on Mac)
flagMacModel	4	Use Mac Model (on Mac)
flagCPUBrandString	8	Use CPU Brand String.
flagWinProductKey	16	Use Product Key (on Windows only)

In Windows 8.1 it looks like `WindowsWMIMBS.InitSecurity(false)` must be called at app.open time as Xojo will do some things in background when opening first window which block our queries.

39.3.46 MacMajorVersion as Integer

Plugin Version: 10.3, Platform: macOS, Targets: All.

Function: The major Mac OS version number.

Example:

```
// show major version number
```

```
MsgBox str(SystemInformationMBS.MacMajorVersion)
```

```
// and show all three version numbers together:
```

```
MsgBox str(SystemInformationMBS.MacMajorVersion)+" "+str(SystemInformationMBS.MacMinorVersion)+" "+str(SystemInformationMBS.MacBugFixVersion)
```

Notes: The major system version number; in 10.4.17 this would be the decimal value 10.

On macOS Big Sur, this returns 11 in Apple Silicon application and 10 in Intel application.

On macOS Monterey this returns 12.

If your Intel application is build with older SDK (e.g. Xojo 2019), the result reported version will still be macOS 10.16.

39.3.47 MacMinorVersion as Integer

Plugin Version: 10.3, Platform: macOS, Targets: All.

Function: The minor Mac OS version number.

Example:

```
MsgBox str(SystemInformationMBS.MacMinorVersion)
```

Notes: The minor system version number; in 10.4.17 this would be the decimal value 4.

39.3.48 MacModel as string

Plugin Version: 8.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the Mac model string.

Example:

```
msgbox SystemInformationMBS.MacModel
```

Notes: for example "<powermac7,3>".

39.3.49 MacROMBootVersion as string

Plugin Version: 19.4, Platform: macOS, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns boot ROM version string.

Example:

```
MsgBox SystemInformationMBS.MacROMBootVersion
```

Notes: e.g. "220.270.99.0.0 (iBridge: 16.16.6571.0.0,0)"

39.3.50 MacSerialNumber as string

Plugin Version: 8.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the serial number of your local Mac.

Example:

```
msgbox SystemInformationMBS.MacSerialNumber
```

Notes: May return an empty string in case of an error.
(e.g. when being user on Windows or if the Mac does not know its serialnumber)

39.3.51 MacUUID as string

Plugin Version: 8.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the unique identifier for the given Mac.

Example:

msgbox SystemInformationMBS.MacUUID

Notes: Returns "" on any error.

39.3.52 MacVRAMSize as Int64

Plugin Version: 8.0, Platform: macOS, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** Queries the size of the main video graphics memory size.

Example:

```
msgbox format(SystemInformationMBS.MacVRAMSize,"0")+ " Bytes VRAM."
```

Notes: Walks over the list of displays, asks on each display for its VRAM size and returns the first value found.

For some reason this seems not to return more than 256 MB of memory.

39.3.53 OSName as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string to display which OS Version you have.

Example:

```
msgBox SystemInformationMBS.OSName  
' may show: Mac OS X
```

Notes: Return values possible:

- Windows NT
- Windows 2000
- Windows XP
- Windows Vista
- Mac OS
- Mac OS X
- Linux

and a few other Windows versions like 6.2 alias 8.

In older plugins this function was named OSNameMBS.

39.3.54 OSVersionString as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a string to display which OS Version you have.

Example:

```
msgBox SystemInformationMBS.OSVersionString  
' may show: Mac OS X 10.1.2
```

Notes: GetOSVersionStr returns the version string of the OS.

Some Possible values:

"System 7.5.3",

"MacOS 8",

"MacOS 9.1",

"MacOS X 10.4.1",

"Windows NT 3.5",

"Windows 2000 (Service Pack 3)".

In older plugins this function was named OSVersionStrMBS.

On macOS Big Sur, this returns 11.x in Apple Silicon application and 10.6 in Intel application.

On macOS Monterey this returns 12.x.

If your Intel application is build with older SDK (e.g. Xojo 2019), the result reported version will still be macOS 10.16.

39.3.55 PhysicalRAM as Double

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the size of the physical installed memory.

Example:

```
msgBox format(SystemInformationMBS.PhysicalRAM/1024/1024,"0")+ " MB of RAM built in."
```

Notes: On old Windows versions some hundred KBs less for DOS.

Result changed from integer to double in plugin version 3.4 to return correct results on 2 GB RAM. In v5.2 changed to return correct values on Macs with more than 4 GB of RAM.

On Windows the total physical memory size.

In older plugins this function was named PhysicalRAMMBS.

39.3.56 ProcessorCount(Mode as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of processors on the target system.

Example:

```
MsgBox _
str(SystemInformationMBS.ProcessorCount(SystemInformationMBS.kProcessorCountDefault ))+" default"
+ EndOfLine + _
str(SystemInformationMBS.ProcessorCount(SystemInformationMBS.kProcessorCountLogical ))+" logical"
+ EndOfLine + _
str(SystemInformationMBS.ProcessorCount(SystemInformationMBS.kProcessorCountPhysical))+" physical"
```

Notes: With plugin version 18.0, we added mode parameter. Pass 2 for physical CPU (kProcessorCountPhysical) and 1 for logical CPU count (kProcessorCountLogical) on Mac/Win.

On Windows or Mac OS the number of cores.

On Linux the number of configured CPUs.

Returns 1 on any error.

39.3.57 ShortUsername as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the short name of the current user.

Example:

```
msgbox "Welcome "+SystemInformationMBS.ShortUsername+"!"
```

Notes: On Linux and Mac OS X the short user name.

On all other cases the same as UserName function.

39.3.58 SystemFont as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the name of the used system font.

Example:

```
msgbox "You are using the system font "+SystemInformationMBS.SystemFont+""
```

Notes: On Windows the system function returns "System" for our tests. Please tell us if this function is not working for you correct on Windows.

On Linux something like "Sans 10" is returned.

In older plugins this function was named SystemFontMBS.

39.3.59 Username as string

Plugin Version: 8.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the name of the current user.

Example:

```
msgbox "Welcome "+SystemInformationMBS.Username+"!"
```

Notes: Code for Multiple User on Mac OS 9 is build in, but I can't test it.

This is tried on Mac OS:

1. Mac OS X Username function.
2. AppleEvent to FileSharing.
3. Multiple User Username (Mac OS 9)
4. System username from system resource.

In older plugins this function was named UserNameMBS.

39.3.60 WinBuildNumber as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: The build number of the operating system.

Example:

```
MsgBox str(SystemInformationMBS.WinBuildNumber)
```

Notes: For example returns 2600 on Windows XP Build 2600.

39.3.61 WinCSDVersion as string

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** A string, such as "Service Pack 3", that indicates the latest Service Pack installed on the system.

Example:

```
MsgBox SystemInformationMBS.WinCSDVersion
```

Notes: Shows here "Service Pack 3" on a Windows XP installation. If no Service Pack has been installed, the string is empty.

39.3.62 WindowsAero as boolean

Plugin Version: 10.1, Platform: Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Whether Windows is using the Aero Interface?

Example:

```
msgbox str(SystemInformationMBS.WindowsAero)
```

Notes: True if aero is used. False if the status is unknown (e.g. on Windows XP) or Aero is not used.

39.3.63 WindowsNativeMachine as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the native machine type.

Notes: This may return ARM64 on a Windows with ARM CPU.

39.3.64 WindowsProcessMachine as Integer

Plugin Version: 22.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the machine type of the current process.

Notes: This may return AMD64, while running in emulation on ARM64.

39.3.65 WinMajorVersion as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: The major version number of the operating system.

Example:

```
MsgBox str(SystemInformationMBS.WinMajorVersion)
```

Notes: Returns 5 on Windows 2000/XP and 6 on Windows Vista/7/8.

39.3.66 WinMinorVersion as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: The minor version number of the operating system.

Example:

```
MsgBox str(SystemInformationMBS.WinMinorVersion)
```

Notes: Returns 0 on Windows 2000/Vista and 1 on Windows XP/7 and 2 on Windows 8.

39.3.67 WinPlatformId as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: The operating system platform.

Example:

```
MsgBox str(SystemInformationMBS.WinPlatformId)
```

Notes: This member can be VER_PLATFORM_WIN32_NT (2).
e.g. 2 on Windows XP.

39.3.68 WinProductKey as string

Plugin Version: 11.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: Returns the product key of the Windows installation.

Notes: Returns "" on any error.

See also:

- 39.3.69 WinProductKey(path as string, name as string, keyStartIndex as Integer = 52) as string 1179

39.3.69 WinProductKey(path as string, name as string, keyStartIndex as Integer = 52) as string

Plugin Version: 11.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: Returns the product key of the Windows installation.

Notes: Returns "" on any error.

With the right path and name you can read also the office versions.

For Office 10, it looks like you need to use 808 as offset.

See also:

- 39.3.68 WinProductKey as string

1178

39.3.70 WinProductType as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: Any additional information about the system.

Example:

MsgBox str(SystemInformationMBS.WinProductType)

Notes: e.g. 1 on Windows XP.

This member can be one of the following values:

VER_NT_DOMAIN_CONTROLLER	2	The system is a domain controller and the operating system is Windows Server 2008, Windows Server 2003, or Windows 2000 Server.
VER_NT_SERVER	3	The operating system is Windows Server 2008, Windows Server 2003, or Windows 2000 Server. Note that a server that is also a domain controller is reported as VER_NT_DOMAIN_CONTROLLER, not VER_NT_SERVER.
VER_NT_WORKSTATION	1	The operating system is Windows Vista, Windows XP Professional, Windows XP Home Edition, or Windows 2000 Professional.

39.3.71 WinServicePackMajor as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** The major version number of the latest Service Pack installed on the system.

Example:

MsgBox str(SystemInformationMBS.WinServicePackMajor)

Notes: For example, for Service Pack 3, the major version number is 3. If no Service Pack has been installed, the value is zero.

39.3.72 WinServicePackMinor as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. **Function:** The minor version number of the latest Service Pack installed on the system.

Example:

```
MsgBox str(SystemInformationMBS.WinServicePackMinor)
```

Notes: For example, for Service Pack 3, the minor version number is 0.

39.3.73 WinSuiteMask as Integer

Plugin Version: 10.3, Platform: Windows, Targets: Desktop, Console & Web.

Function: A bit mask that identifies the product suites available on the system.

Example:

```
MsgBox str(SystemInformationMBS.WinSuiteMask)
```

Notes: e.g. 256 on Windows XP.

This member can be a combination of the following values.

39.3.74 Constants

ProcessorCount Modes

Constant	Value	Description
kProcessorCountDefault	0	Default behavior.
kProcessorCountLogical	1	Logical Processors
kProcessorCountPhysical	2	Physical Processors

Windows Machine Types

VER_SUITE_BACKOFFICE	4	Microsoft BackOffice components are installed.
VER_SUITE_BLADE	&h0400	Windows Server 2003, Web Edition is installed.
VER_SUITE_COMPUTE_SERVER	&h4000	Windows Server 2003, Compute Cluster Edition is installed.
VER_SUITE_DATACENTER	&h0080	Windows Server 2008 Datacenter, Windows Server 2003, Datacenter Edition, or Windows 2000 Datacenter Server is installed.
VER_SUITE_ENTERPRISE	2	Windows Server 2008 Enterprise, Windows Server 2003, Enterprise Edition, or Windows 2000 Advanced Server is installed. Refer to the Remarks section for more information about this bit flag.
VER_SUITE_EMBEDDEDNT	&h0040	Windows XP Embedded is installed.
VER_SUITE_PERSONAL	&h0200	Windows Vista Home Premium, Windows Vista Home Basic, or Windows XP Home Edition is installed.
VER_SUITE_SINGLEUSERTS	&h0100	Remote Desktop is supported, but only one interactive session is supported. This value is set unless the system is running in application server mode.
VER_SUITE_SMALLBUSINESS	1	Microsoft Small Business Server was once installed on the system, but may have been upgraded to another version of Windows. Refer to the Remarks section for more information about this bit flag.
VER_SUITE_SMALLBUSINESS_RESTRICTED	&h0020	Microsoft Small Business Server is installed with the restrictive client license in force. Refer to the Remarks section for more information about this bit flag.
VER_SUITE_STORAGE_SERVER	&h2000	Windows Storage Server 2003 R2 or Windows Storage Server 2003 is installed.
VER_SUITE_TERMINAL	&h0010	Terminal Services is installed. This value is always set. If VER_SUITE_TERMINAL is set but VER_SUITE_SINGLEUSERTS is not set, the system is running in application server mode.
VER_SUITE_WH_SERVER	&h8000	Windows Home Server is installed.

Constant	Value	Description
kWindowsMachineAMD64	&h8664	ARM in 64-bit.
kWindowsMachineARM32	&h01c0	ARM in 32-bit.
kWindowsMachineARM64	&hAA64	AMD 64-bit x86.
kWindowsMachineI386	&h014c	Intel 32-bit x86.
kWindowsMachineUnknown	0	Unknown.

Chapter 40

TimeZone

40.1 class TimeZoneMBS

40.1.1 class TimeZoneMBS

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Deprecated: This item is deprecated and should no longer be used. You can use TimeZone instead.

Function: Allows you to access information about the current time zone.

Example:

```
Var t as new TimeZoneMBS
MsgBox str(t.GmtDeltaTotalseconds)
```

Notes: on Mac OS Cocoa applications, better use newer NSTimezoneMBS class.

Blog Entries

- [Cleanup Xojo Plugins](#)
- [MBS Xojo Plugins, version 20.2pr7](#)
- [MBS Xojo / Real Studio Plugins, version 14.2pr5](#)

40.1.2 Properties

40.1.3 DaylightName as String

Plugin Version: 14.2, Platform: Windows, Targets: Desktop, Console & Web.

Function: A description for daylight saving time.

Example:

```
Var t as new TimeZoneMBS
MsgBox t.DaylightName
```

Notes: For example, "PDT" could indicate Pacific Daylight Time. This string can be empty.
 e.g. "Mittleeuropäische Sommerzeit"
 On Mac, please use NSTimeZoneMBS class.
 (Read only property)

40.1.4 GmtDeltaHours as Integer

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the GMT offset in hours.

Example:

```
Var t as TimeZoneMBS
```

```
t=new TimeZoneMBS
```

```
msgbox format(t.GmtDeltaHours,"+00")+ ":" +format(t.GmtDeltaminutes,"00")+ ":" +format(t.GmtDeltaseconds,"00")
```

Notes: On Windows this information is not available.
 (Read only property)

40.1.5 GmtDeltaMinutes as Integer

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the GMT offset in minutes.

Example:

```
Var t as TimeZoneMBS
```

```
t=new TimeZoneMBS
```

```
msgbox format(t.GmtDeltaHours,"+00")+ ":" +format(t.GmtDeltaminutes,"00")+ ":" +format(t.GmtDeltaseconds,"00")
```

Notes: On Windows this information is not available.
 (Read only property)

40.1.6 GmtDeltaSeconds as Integer

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the GMT offset in seconds.

Example:

```
Var t as TimeZoneMBS
```

```
t=new TimeZoneMBS
```

```
msgbox format(t.GmtDeltaHours,"+00")+ ":" +format(t.GmtDeltaminutes,"00")+ ":" +format(t.GmtDeltasec-
onds,"00")
```

Notes: (Read only property)

40.1.7 GmtDeltaTotalseconds as Integer

Platforms: macOS, Windows, Targets: Desktop, Console & Web.

Function: Returns the GMT offset in seconds.

Example:

```
Var t as new TimeZoneMBS
```

```
MsgBox str(t.GmtDeltaTotalseconds)
```

Notes: (Read only property)

40.1.8 Latitude as Double

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the latitude of the current position on earth.

Example:

```
Var longdeg,longmin as Integer
```

```
Var t as TimeZoneMBS
```

```
t=new TimeZoneMBS
```

```
longdeg=floor(t.longitude*90)
```

```
longmin=floor((t.longitude*90-longdeg)*60)
```

```
statictext1.text=format(longdeg,"0")+ "°-∞"
```

```
statictext2.text=format(longmin,"0")+""
```

Notes: On Windows this information is not available.
(Read only property)

40.1.9 Longitude as Double

Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the longitude of the current position on earth.

Example:

```
Var latdeg,latmin as Integer
Var t as TimeZoneMBS
```

```
t=new TimeZoneMBS
```

```
latdeg=floor(t.latitude*90)
latmin=floor((t.latitude*90-latdeg)*60)
```

```
statictext1.text=format(latdeg,"0")+"-∞"
statictext2.text=format(latmin,"0")+""
```

Notes: On Windows this information is not available.
(Read only property)

40.1.10 StandardName as String

Plugin Version: 14.2, Platform: Windows, Targets: Desktop, Console & Web.

Function: A description for standard time.

Example:

```
Var t as new TimeZoneMBS
MsgBox t.StandardName
```

Notes: For example, "EST" could indicate Eastern Standard Time. This string can be empty.
e.g. "Mitteleuropäische Zeit"
On Mac, please use NSTimeZoneMBS class.
(Read only property)

Chapter 41

Window

41.1 class DesktopWindow

41.1.1 class DesktopWindow

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Extends Xojo's Window Class.

Example:

```
window1.HasNoTitleBarMBS = true
```

Notes: In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

41.1.2 Methods

41.1.3 ActivateWindowMBS

Plugin Version: 21.5, Platforms: macOS, Windows, Targets: Desktop only.

Function: Activates the window.

Notes: This function does more than just show. if the window has been minimized, it will restore it. Also the window is moved to front and made the current window for keyboard input.

Sadly windows limits which windows can go to front, so some windows may not come to front unless they are clicked on.

As activation can be asynchronously, the window may activate later, e.g. after current method ends.

For MacOS Cocoa apps, we simply bring the window to the front.

41.1.4 BackingScaleFactorMBS as double

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns the scale factor representing the number of backing store pixels corresponding to each linear unit in window space on this window.

Notes: This is generally only necessary when building a bitmap context or image whose resolution needs to match that of a particular Window. Note that a Window's backing scale factor can change over time, such as when the window moves from one display to another, or when a display's resolution changes, so clients should not cache the value returned by this function.

If platform does not support scaling factor, we return 1.0. (on Windows, Linux and older Mac OS X) Supported for Carbon and Cocoa windows.

For apps which are not enabled for retina support, the function returns 1. So you only see 2 here if app is Cocoa, display is retina and info.plist has the NSHighResolutionCapable key.

41.1.5 CleanUpTransparentMBS(refValue as integer)

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Cleans up transparency support for a Xojo window.

Example:

```
Var p as Integer // property on Windows
```

```
p = window1.MakeTransparentMBS
```

```
// later in close event
```

```
window1.CleanUpTransparentMBS p
```

Notes: Call this in the close event of a window passing the value you got from the MakeTransparentMBS call.

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel.

41.1.6 ClearTransparencyMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Clears transparency of a window.

41.1.7 CollapsibleMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns true if this window is collapsed. You can set it.

Example:

```
mainwindow.collapsedMBS=false 'show window
```

Notes: Collapseable windows are the normal document windows.

You can't collapse dialogs or floating windows well.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

41.1.8 ConstrainWindowToScreenMBS(animate as Boolean)

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Makes sure the window is on one screen visible.

Notes: Implemented on Mac for both Carbon and Cocoa.

41.1.9 IsFullScreenMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Queries whether window is fullscreen.

Notes: Indicates that a window has fullscreen appearance.

A fullscreen window does not draw its titlebar, and may have special handling for its toolbar.

Available in Mac OS X v10.7 and later.

41.1.10 MakeTransparentMBS as integer

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Installs transparency support to a Xojo window.

Example:

```
Var p as Integer // property on Windows
```

```
p = window1.MakeTransparentMBS
```

```
// later in close event
```

```
window1.CleanUpTransparentMBS p
```

Notes: Will return non zero value if successful.
The window doesn't change if you don't use the CGContextMBS property.

Requires Mac OS X, Windows 2000 or Windows XP to work. The window must be a document window.

Set MacProcID of the window to 1104 and you can make it transparent without a title bar. (on Mac OS X)

With plugin version 11.1, we now return a value which should store with the window. Later in the Close event, you call CleanUpTransparentMBS passing this value.

Added Cocoa support in 11.3 plugins.

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel. Calls to MakeTransparentMBS and CleanUpTransparentMBS are not required for Linux.

41.1.11 RefreshThreadSafeMBS(immediately As Boolean = False)

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Refresh the rectangle.

Notes: This method is to allow you to refresh in a thread without a problem.

If called on main thread, the plugin will simply call Refresh method directly.

If called on other threads the plugin will schedule to call Refresh method a short time later on the main thread.

See also:

- 41.1.12 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately As Boolean = False) 1190

41.1.12 RefreshThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, immediately As Boolean = False)

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Refresh the rectangle.

Notes: This method is to allow you to refresh in a thread without a problem.

If called on main thread, the plugin will simply call RefreshRect method directly.

If called on other threads the plugin will schedule to call RefreshRect method a short time later on the main thread.

See also:

- 41.1.11 RefreshThreadSafeMBS(immediately As Boolean = False)

1190

41.1.13 RemoveWindowProxyIconMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Removes any Proxy Icon from the Window.

Example:

```
mainwindow.RemoveWindowProxyIconMBS
```

Notes: Requires Mac OS 8.5 or newer.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

41.1.14 SetTransparencyMBS(value as integer) as boolean

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Your window can have transparency on Mac OS X.

Example:

```
if window1.SetTransparencyMBS(127) then
  'fine
else
  'error
end if
```

Notes: Use values between 0 and 255.

Requires Mac OS X, Windows 2000 or Windows XP to work.

May be limited to simple windows like normal document windows.

You need to call MakeTransparent before to install transparency.

(this function was replaced in v4.4)

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel.

41.1.15 ShowHideToolbarMBS(animate as Boolean, value as Boolean)

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Shows or hides the toolbar.

Notes: Only working on MacOS.

41.1.16 SmoothResizeCenteredMBS(width as integer,height as integer)

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Resizes the window smoothly to all directions.

Example:

```
Var w,h as Integer
```

```
// get destination dimension
```

```
w=300
```

```
h=200
```

```
// Resize
```

```
window1.SmoothResizeCenteredMBS w,h
```

```
// Now resize via RB to make it permanent.
```

```
width=w
```

```
height=h
```

Notes: You need to set the new size using Xojo's Windows properties, too. (no longer needed with Xojo) In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

41.1.17 SmoothResizeMBS(width as integer,height as integer)

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Resizes the window smoothly.

Example:

```
Var w,h as Integer
```

```
// get destination dimension
```

```
w=300
```



```
h=200
```

```
// Resize
```

```
window1.SmoothResizeMBS w,h
```

```
// Now resize via RB to make it permanent.
```

```
width=w
```

```
height=h
```

Notes: You need to set the new size using Xojo's Windows properties, too. (no longer needed with Xojo) In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

41.1.18 ToggleFullScreenMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Toggles fullscreen mode for this window.

Notes: Returns true on success.

If an application supports fullscreen, it should add a menu item to the View menu with toggleFullScreen as the action.

Available in Mac OS X v10.7 and later.

This method does not much if you don't mark a window to be the primary fullscreen window.

41.1.19 WinFlashWindowMBS(Invert as boolean)

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Flashes Window.

Example:

```
window1.WinFlashWindowMBS true
```

Notes: Flashes the specified window one time. It does not change the active state of the window.

Invert: If this parameter is true, the window is flashed from one state to the other. If it is false, the window is returned to its original state (either active or inactive).

When an application is minimized and this parameter is true, the taskbar window button flashes active/inactive. If it is false, the taskbar window button flashes inactive, meaning that it does not change colors. It flashes, as if it were being redrawn, but it does not provide the visual invert clue to the user.

Flashing a window means changing the appearance of its caption bar as if the window were changing from inactive to active status, or vice versa. (An inactive caption bar changes to an active caption bar; an active caption bar changes to an inactive caption bar.)

Typically, a window is flashed to inform the user that the window requires attention but that it does not currently have the keyboard focus.

The `FlashWindow` function flashes the window only once; for repeated flashing, the application should create a system timer.

41.1.20 Properties

41.1.21 `CanBeVisibleWithoutLoginMBS` as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window can be visible before login.

Example:

```
window1.CanBeVisibleWithoutLoginMBS = true
```

Notes: This window can be made visible prior to user login. By default, in Mac OS X 10.5 and later no windows can be visible before a user logs into the system; this protects the user against certain types of malicious use of insecure applications. However, some software, such as input methods or other accessibility software, may need to deliberately make windows available prior to user login. Such software should add this window attribute to its windows. Available for all windows in Mac OS X 10.5 and later.

(Read and Write computed property)

41.1.22 `CollapsedMBS` as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns true if this window can be collapsed.

Example:

```
MsgBox str(window1.CollapsibleMBS)
```

Notes: Collapseable windows are the normal document windows.

You can't collapse dialogs or floating windows well.

In Xojo 2005 and newer you need to use `self.` in front of the method as the property name alone is not accepted.

(Read and Write computed property)

41.1.23 FullScreenAuxiliaryMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window is setup as being the auxiliary fullscreen window.

Notes: (Read and Write computed property)

41.1.24 FullScreenPrimaryMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window is setup as being the primary fullscreen window.

Notes: If true, the window receives the fullscreen widget in the title bar.

(Read and Write computed property)

41.1.25 HasBorderMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Decides whether the window has a border.

Example:

```
mainwindow.HasborderMBS=false ' Remove border
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it). This option removes the title bar. HasCaptionMBS removes even more.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.26 HasCaptionMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Whether the window has a caption (a title bar)

Example:

```
Window1.HasCaptionMBS=false
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

(Read and Write computed property)

41.1.27 HasCloseBoxMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window has a CloseBox.

Example:

```
mainwindow.HasCloseboxMBS=false 'remove closebox
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

(Read and Write computed property)

41.1.28 HasCollapseBoxMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window has a Collapsebox.

Example:

```
mainwindow.HasCollapseBoxMBS=false 'remove Collapsebox
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

(Read and Write computed property)

41.1.29 HasMaximizeBoxMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Returns true if this window has a Maximize Button.

Example:

```
mainwindow.HasMaximizeBoxMBS=false 'remove Maximize button
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.30 HasMinimizeBoxMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Returns true if this window has a Minimize Button.

Example:

```
mainwindow.HasMinimizeBoxMBS=false 'remove Minimize button
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.31 HasNoShadowMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window has no shadow.

Example:

```
mainwindow.HasNoShadowMBS=true 'remove shadow
```

Notes: Available for all windows on Mac OS X. This attribute is automatically given to windows of kOverlayWindowClass.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

(Read and Write computed property)

41.1.32 HasNoTitleBarMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use `NSWindowMBS` for Cocoa instead. **Function:** For Document, Floating, and Utility windows, this attribute allows you to hide the title bar of a window.

Example:

```
window1.HasNoTitleBarMBS = true
```

// for Cocoa:

```
window1.HasCloseBoxMBS = false  
window1.HasCollapseBoxMBS = false  
window1.IsResizableMBS = false  
window1.HasNoTitleBarMBS = true
```

Notes: For Mac OS X 10.4 or later.

You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

(Read and Write computed property)

41.1.33 HasSystemMenuMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Whether the window has a system menu inside the title bar.

Example:

```
Window1.HasSystemMenuMBS=false
```

Notes: This property has only an effect on Windows. It will disable the system menu and also the minimize, maximize and close buttons. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.34 HasToolBarButtonMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window has a toolbar button.

Example:

```
mainwindow.HasToolBarButtonMBS=true 'shows Toolbar Button
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon. The Button is shown on next redraw of the window frame.

Use the `CarbonWindowsEventsMBS` class to receive events when the button is pressed.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.35 IgnoreClicksMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Can be used to control whether mouse clicks are ignored for this window.

Example:

```
Window1.IgnoreClicksMBS=true
```

Notes: Whether this window never receives mouse events, even in areas that are opaque. Instead, clicks on the window will be passed through to windows beneath it. Available for all windows on Mac OS X 10.2 and later.

You can read the state on Mac Classic, but you can only change it on Mac Carbon.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.36 IsIconicMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Returns true if this window is inside the taskbar.

Example:

```
MsgBox str(window1.IsIconicMBS)
```

Notes: If you set IsIconic to true the window is minimized and if you set it to false the window size and position is restored.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.37 IsMetalWindowMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window uses the Metal appearance.

Example:

```
if window1.Frame = window.FrameTypeMetal then
  MsgBox "Window is metal. "+str(window1.IsMetalWindowMBS)
else
  MsgBox "Window is not metal. "+str(window1.IsMetalWindowMBS)
end if
```

Notes: Available for document windows on Mac OS X 10.2 and later, and for floating windows on Mac OS X 10.3 and later. Drawers can also be metal, but dynamically adjust their appearance based on their parent window's appearance; it is not necessary to specify this attribute for a metal drawer.

You can read the state on Mac Classic, but you can only change it on Mac Carbon.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.38 IsResizableMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window is resizable.

Example:

```
Window1.IsResizableMBS=false
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

Use GrowBoxTransparentMBS on Composite Mac OS windows to enable the transparent grow box.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Works in Cocoa.

(Read and Write computed property)

41.1.39 IsZoomedMacMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns true if the window is zoomed.

Example:

```
MsgBox str(window1.IsZoomedMacMBS)
```

Notes: Requires Mac OS 8.5 or newer.

Seems not to work correctly on RB 5.5.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Added Cocoa support in plugin version 10.0.

(Read and Write computed property)

41.1.40 IsZoomedMBS as Boolean

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: Returns true if this window has been maximized.

Example:

```
MsgBox str(window1.IsZoomedMBS)
```

Notes: If you set `IsZoomed` to true the window is maximized and if you set it to false the window size and position is restored.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

Works on Cocoa.

(Read and Write computed property)

41.1.41 ModifiedMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: You can set or get the value of the modified state.

Example:

```
mainwindow.modifiedMBS=true
```

Notes: Requires Mac OS 8.5 or newer.

As long as you don't set `modified` to false the window keeps to tell you that it's modified. Not sure why, so just set `modified` to false early in creating the window.

(Read and Write computed property)

41.1.42 ToolbarVisibleMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether the toolbar is shown in this window or not.

Notes: Value is false on any error.

Only working on MacOS.

(Read and Write computed property)

41.1.43 TransparencyMBS as single

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The transparency of the window on Mac OS X, Windows 2000 and Windows XP.

Notes: 1 for opaque, 0 for invisible.

Return 1 on any error. On Windows it returns always 1 as the current transparency value can't be queried. You need to call `MakeTransparent` before to install transparency.

(Read and Write computed property)

41.1.44 UnifiedTitleAndToolbarMBS as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether this window has an unified title and toolbar look.

Example:

```
window1.UnifiedTitleAndToolbarMBS = true
```

Notes: This window draws its window title and toolbar using a unified appearance that has no separator between the two areas. A window may not have both UnifiedTitleAndToolbar and Metal appearance. If a window already has the metal attribute, attempting to set the Unified attribute will cause ChangeWindows to return an error, and vice versa. This constant was not added to this header file until Mac OS X 10.5, but it is actually available at runtime on Mac OS X 10.4 and later for windows of kDocumentWindowClass. However, on Mac OS X 10.5 and later, kHIWindowBitUnifiedTitleAndToolbar no longer has any effect, since all windows with toolbars now have a unified look.

(Read and Write computed property)

41.1.45 WindowProxyIconFileMBS as folderitem

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: That's the icon to the window which belongs to the file.

Example:

```
Var f as folderItem
```

```
f=getopenFolderItem("special/any")
if f<>nil then
window1.WindowProxyIconFileMBS=f
end if
```

Notes: Requires Mac OS 8.5 or newer.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.1.46 WinMenuHandleMBS as integer

Plugin Version: 21.5, Platform: Windows, Targets: Desktop only.

Function: A property to access the handle used for the menu of a Window.

Example:

```
Var menu as Integer // global
```

```
if menu=0 then
menu=Window1.WinMenuHandleMBS // read it on the first window
else
Window1.winmenuHandleMBS=menu // set it on the second window
end if
```

Notes: Used in the example "Menu in every Window" to have a menubar in every window on Windows. But never forget to quit your app after all windows were closed.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2 class Window

41.2.1 class Window

Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Extends Xojo's Window Class.

Example:

```
window1.HasNoTitleBarMBS = true
```

Notes: In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

41.2.2 Methods

41.2.3 ActivateWindowMBS

Plugin Version: 19.2, Platforms: macOS, Windows, Targets: Desktop only.

Function: Activates the window.

Notes: This function does more than just show. if the window has been minimized, it will restore it. Also the window is moved to front and made the current window for keyboard input.

Sadly windows limits which windows can go to front, so some windows may not come to front unless they are clicked on.

As activation can be asynchronously, the window may activate later, e.g. after current method ends.

For MacOS Cocoa apps, we simply bring the window to the front.

Blog Entries

- [MBS Xojo Plugins in version 19.2](#)
- [MBS Xojo Plugins, version 19.2pr1](#)
- [SetFrontMost for Windows](#)

Videos

- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [17.5, page 42: What's New in the MBS Plugins, With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes](#)

41.2.4 BackingScaleFactorMBS as Double

Plugin Version: 12.3, Platform: macOS, Targets: Desktop only.

Function: Returns the scale factor representing the number of backing store pixels corresponding to each linear unit in window space on this window.

Notes: This is generally only necessary when building a bitmap context or image whose resolution needs to match that of a particular Window. Note that a Window's backing scale factor can change over time, such as when the window moves from one display to another, or when a display's resolution changes, so clients should not cache the value returned by this function.

If platform does not support scaling factor, we return 1.0. (on Windows, Linux and older Mac OS X) Supported for Carbon and Cocoa windows.

For apps which are not enabled for retina support, the function returns 1. So you only see 2 here if app is Cocoa, display is retina and info.plist has the NSHighResolutionCapable key.

Blog Entries

- [MBS Real Studio Plugins, version 12.4pr1](#)
- [MBS Real Studio Plugins, version 12.3pr13](#)
- [High Resolution is coming](#)
- [MBS Real Studio Plugins, version 12.3pr5](#)

41.2.5 CleanUpTransparentMBS(refValue as Integer)

Plugin Version: 5.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Cleans up transparency support for a Xojo window.

Example:

```
Var p as Integer // property on Windows
```

```
p = window1.MakeTransparentMBS
```

```
// later in close event
```

```
window1.CleanUpTransparentMBS p
```

Notes: Call this in the close event of a window passing the value you got from the MakeTransparentMBS call.

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel.

Blog Entries

- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr5](#)

41.2.6 ClearTransparencyMBS

Plugin Version: 5.0, Platform: macOS, Targets: Desktop only.

Function: Clears transparency of a window.

41.2.7 CollapsibleMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: Returns true if this window can be collapsed.

Example:

```
MsgBox str(window1.CollapsibleMBS)
```

Notes: Collapseable windows are the normal document windows.

You can't collapse dialogs or floating windows well.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Added Cocoa support in plugin version 10.0.

41.2.8 ConstrainWindowToScreenMBS(animate as boolean)

Plugin Version: 10.5, Platform: macOS, Targets: Desktop only.

Function: Makes sure the window is on one screen visible.

Notes: Implemented on Mac for both Carbon and Cocoa.

Blog Entries

- [MBS REALbasic Plugins, version 10.6pr6](#)

41.2.9 InvalidateRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)

Plugin Version: 19.5, Platform: macOS, Targets: Desktop only.

Function: Invalidate the rectangle.

Notes: This method is to allow you to invalidate in a thread without a problem.

If called on main thread, the plugin will simply call `InvalidateRect` method directly.

If called on other threads the plugin will schedule to call `InvalidateRect` method a short time later on the main thread.

Blog Entries

- [MBS Xojo Plugins, version 19.5pr3](#)

41.2.10 `InvalidateThreadSafeMBS(EraseBackground as boolean = true)`

Plugin Version: 19.5, Platform: macOS, Targets: Desktop only.

Function: Invalidate the window.

Notes: This method is to allow you to invalidate in a thread without a problem.

If called on main thread, the plugin will simply call `Invalidate` method directly.

If called on other threads the plugin will schedule to call `Invalidate` method a short time later on the main thread.

Blog Entries

- [MBS Xojo Plugins, version 20.0pr6](#)
- [MBS Xojo Plugins, version 19.5pr3](#)
- [MBS Real Studio Plugins, version 13.0pr5](#)

41.2.11 `IsFullScreenMBS as Boolean`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Queries whether window is fullscreen.

Notes: Indicates that a window has fullscreen appearance.

A fullscreen window does not draw its titlebar, and may have special handling for its toolbar.

Available in Mac OS X v10.7 and later.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr6](#)
- [Lion features for Real Studio](#)
- [MBS Real Studio Plugins, version 11.2pr11](#)
- [MBS Real Studio Plugins, version 11.2pr9](#)

41.2.12 MakeTransparentMBS as Integer

Plugin Version: 5.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Installs transparency support to a Xojo window.

Example:

```
Var p as Integer // property on Windows
```

```
p = window1.MakeTransparentMBS
```

```
// later in close event
```

```
window1.CleanUpTransparentMBS p
```

Notes: Will return non zero value if successful.

The window doesn't change if you don't use the CGContextMBS property.

Requires Mac OS X, Windows 2000 or Windows XP to work. The window must be a document window.

Set MacProcID of the window to 1104 and you can make it transparent without a title bar. (on Mac OS X)

With plugin version 11.1, we now return a value which should store with the window. Later in the Close event, you call CleanUpTransparentMBS passing this value.

Added Cocoa support in 11.3 plugins.

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel. Calls to MakeTransparentMBS and CleanUpTransparentMBS are not required for Linux.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr3](#)
- [MBS REALbasic Plugins, version 11.2pr1](#)
- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr5](#)

Xojo Developer Magazine

- [10.1, page 76: Using Plugins, Working with the Monkeybread Plugins by Marc Zeedar](#)

41.2.13 RefreshRectThreadSafeMBS(X As Integer, Y As Integer, Width As Integer, Height As Integer, EraseBackground as boolean = true)

Plugin Version: 19.5, Platform: macOS, Targets: Desktop only.

Function: Refresh the rectangle.

Notes: This method is to allow you to refresh in a thread without a problem.

If called on main thread, the plugin will simply call RefreshRect method directly.

If called on other threads the plugin will schedule to call RefreshRect method a short time later on the main thread.

Blog Entries

- [MBS Xojo Plugins, version 19.5pr3](#)

41.2.14 RefreshThreadSafeMBS(EraseBackground as boolean = true)

Plugin Version: 19.5, Platform: macOS, Targets: Desktop only.

Function: Refresh the rectangle.

Notes: This method is to allow you to refresh in a thread without a problem.

If called on main thread, the plugin will simply call Refresh method directly.

If called on other threads the plugin will schedule to call Refresh method a short time later on the main thread.

Blog Entries

- [MBS Xojo Plugins, version 19.5pr3](#)
- [MBS Real Studio Plugins, version 13.0pr5](#)

41.2.15 RemoveWindowProxyIconMBS

Platform: macOS, Targets: Desktop only.

Function: Removes any Proxy Icon from the Window.

Example:

```
mainwindow.RemoveWindowProxyIconMBS
```

Notes: Requires Mac OS 8.5 or newer.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Blog Entries

- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr4](#)

41.2.16 SetTransparencyMBS(value as Integer) as boolean

Plugin Version: 5.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Your window can have transparency on Mac OS X.

Example:

```
if window1.SetTransparencyMBS(127) then
'fine
else
'error
end if
```

Notes: Use values between 0 and 255.

Requires Mac OS X, Windows 2000 or Windows XP to work.

May be limited to simple windows like normal document windows.

You need to call MakeTransparent before to install transparency.

(this function was replaced in v4.4)

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel.

41.2.17 ShowHideToolbarMBS(animate as boolean, value as boolean)

Plugin Version: 4.0, Platform: macOS, Targets: Desktop only.

Function: Shows or hides the toolbar.

Notes: Only working on MacOS.

41.2.18 SmoothResizeCenteredMBS(width as Integer,height as Integer)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Resizes the window smoothly to all directions.

Example:

```
Var w,h as Integer
```

```
// get destination dimension
```

```
w=300
```

```
h=200
```

```
// Resize
```

```
window1.SmoothResizeCenteredMBS w,h
```

```
// Now resize via RB to make it permanent.
```

```
width=w  
height=h
```

Notes: You need to set the new size using Xojo's Windows properties, too. (no longer needed with Xojo) In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr3](#)

41.2.19 SmoothResizeMBS(width as Integer,height as Integer)

Platform: macOS, Targets: Desktop only.

Function: Resizes the window smoothly.

Example:

```
Var w,h as Integer
```

```
// get destination dimension
```

```
w=300  
h=200
```

```
// Resize
```

```
window1.SmoothResizeMBS w,h
```

```
// Now resize via RB to make it permanent.
```

```
width=w  
height=h
```

Notes: You need to set the new size using Xojo's Windows properties, too. (no longer needed with Xojo) In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Works on Cocoa on 10.4 and newer.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr13](#)

41.2.20 ToggleFullScreenMBS as Boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Toggles fullscreen mode for this window.

Notes: Returns true on success.

If an application supports fullscreen, it should add a menu item to the View menu with toggleFullScreen as the action.

Available in Mac OS X v10.7 and later.

This method does not much if you don't mark a window to be the primary fullscreen window.

Blog Entries

- [Lion features for Real Studio](#)
- [MBS Real Studio Plugins, version 11.2pr11](#)
- [MBS Real Studio Plugins, version 11.2pr9](#)

41.2.21 WinFlashWindowMBS(Invert as boolean)

Plugin Version: 16.2, Platform: Windows, Targets: Desktop only.

Function: Flashes Window.

Example:

```
window1.WinFlashWindowMBS true
```

Notes: Flashes the specified window one time. It does not change the active state of the window.

Invert: If this parameter is true, the window is flashed from one state to the other. If it is false, the window is returned to its original state (either active or inactive).

When an application is minimized and this parameter is true, the taskbar window button flashes active/inactive. If it is false, the taskbar window button flashes inactive, meaning that it does not change colors. It flashes, as if it were being redrawn, but it does not provide the visual invert clue to the user.

Flashing a window means changing the appearance of its caption bar as if the window were changing from inactive to active status, or vice versa. (An inactive caption bar changes to an active caption bar; an active caption bar changes to an inactive caption bar.)

Typically, a window is flashed to inform the user that the window requires attention but that it does not currently have the keyboard focus.

The FlashWindow function flashes the window only once; for repeated flashing, the application should create a system timer.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.2pr2](#)

41.2.22 Properties

41.2.23 CanBeVisibleWithoutLoginMBS as Boolean

Plugin Version: 8.1, Platform: macOS, Targets: Desktop only.

Function: Whether this window can be visible before login.

Example:

```
window1.CanBeVisibleWithoutLoginMBS = true
```

Notes: This window can be made visible prior to user login. By default, in Mac OS X 10.5 and later no windows can be visible before a user logs into the system; this protects the user against certain types of malicious use of insecure applications. However, some software, such as input methods or other accessibility software, may need to deliberately make windows available prior to user login. Such software should add this window attribute to its windows. Available for all windows in Mac OS X 10.5 and later.
(Read and Write computed property)

41.2.24 collapsedMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: Returns true if this window is collapsed. You can set it.

Example:

```
mainwindow.collapsedMBS=false 'show window
```

Notes: Collapseable windows are the normal document windows.

You can't collapse dialogs or floating windows well.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Added Cocoa support in plugin version 10.0.

(Read and Write computed property)

41.2.25 FullScreenAuxiliaryMBS as Boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Whether this window is setup as being the auxiliary fullscreen window.

Notes: (Read and Write computed property)

41.2.26 FullScreenPrimaryMBS as Boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Whether this window is setup as being the primary fullscreen window.

Notes: If true, the window receives the fullscreen widget in the title bar.

(Read and Write computed property)

41.2.27 HasborderMBS as boolean

Platform: Windows, Targets: Desktop only.

Function: Decides whether the window has a border.

Example:

```
mainwindow.HasborderMBS=false ' Remove border
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it). This option removes the title bar. HasCaptionMBS removes even more.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.28 HasCaptionMBS as Boolean

Plugin Version: 7.4, Platform: Windows, Targets: Desktop only.

Function: Whether the window has a caption (a title bar)

Example:

```
Window1.HasCaptionMBS=false
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.29 HasCloseBoxMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: Whether this window has a CloseBox.

Example:

```
mainwindow.HasCloseboxMBS=false 'remove closebox
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.30 HasCollapseBoxMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: Whether this window has a Collapsebox.

Example:

```
mainwindow.HasCollapseBoxMBS=false 'remove Collapsebox
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.31 HasMaximizeBoxMBS as boolean

Platform: Windows, Targets: Desktop only.

Function: Returns true if this window has a Maximize Button.

Example:

```
mainwindow.HasMaximizeBoxMBS=false 'remove Maximize button
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.32 HasMinimizeBoxMBS as boolean

Platform: Windows, Targets: Desktop only.

Function: Returns true if this window has a Minimize Button.

Example:

```
mainwindow.HasMinimizeBoxMBS=false 'remove Minimize button
```

Notes: This property has only an effect on Windows. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.33 HasNoShadowMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: Whether this window has no shadow.

Example:

```
mainwindow.HasNoShadowMBS=true 'remove shadow
```

Notes: Available for all windows on Mac OS X. This attribute is automatically given to windows of `kOverlayWindowClass`.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

Added Cocoa support in plugin version 10.0.

(Read and Write computed property)

41.2.34 HasNoTitleBarMBS as Boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use `NSWindowMBS` for Cocoa instead. **Function:** For Document, Floating, and Utility windows, this attribute allows you to hide the title bar of a window.

Example:

```
window1.HasNoTitleBarMBS = true
```

// for Cocoa:

```
window1.HasCloseBoxMBS = false  
window1.HasCollapseBoxMBS = false  
window1.IsResizableMBS = false  
window1.HasNoTitleBarMBS = true
```

Notes: For Mac OS X 10.4 or later.

You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

Added Cocoa support in Plugin version 10.4.
(Read and Write computed property)

41.2.35 HasSystemMenuMBS as Boolean

Plugin Version: 7.4, Platform: Windows, Targets: Desktop only.

Function: Whether the window has a system menu inside the title bar.

Example:

```
Window1.HasSystemMenuMBS=false
```

Notes: This property has only an effect on Windows. It will disable the system menu and also the minimize, maximize and close buttons. The window needs to be redrawn before the change is visible (for example move it).

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.
(Read and Write computed property)

41.2.36 HasToolBarButtonMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: Whether this window has a toolbar button.

Example:

```
mainwindow.HasToolBarButtonMBS=true 'shows Toolbar Button
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon. The Button is shown on next redraw of the window frame.

Use the CarbonWindowsEventsMBS class to receive events when the button is pressed.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

(Read and Write computed property)

41.2.37 IgnoreClicksMBS as Boolean

Plugin Version: 3.2, Platform: macOS, Targets: Desktop only.

Function: Can be used to control whether mouse clicks are ignored for this window.

Example:

```
Window1.IgnoreClicksMBS=true
```

Notes: Whether this window never receives mouse events, even in areas that are opaque. Instead, clicks on the window will be passed through to windows beneath it. Available for all windows on Mac OS X 10.2 and later.

You can read the state on Mac Classic, but you can only change it on Mac Carbon.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

Added Cocoa support in plugin version 10.0.

(Read and Write computed property)

41.2.38 IsIconicMBS as boolean

Platform: Windows, Targets: Desktop only.

Function: Returns true if this window is inside the taskbar.

Example:

```
MsgBox str(window1.IsIconicMBS)
```

Notes: If you set IsIconic to true the window is minimized and if you set it to false the window size and position is restored.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.39 IsMetalWindowMBS as Boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Whether this window uses the Metal appearance.

Example:

```
if window1.Frame = window.FrameTypeMetal then
  MsgBox "Window is metal. "+str(window1.IsMetalWindowMBS)
else
  MsgBox "Window is not metal. "+str(window1.IsMetalWindowMBS)
end if
```

Notes: Available for document windows on Mac OS X 10.2 and later, and for floating windows on Mac OS X 10.3 and later. Drawers can also be metal, but dynamically adjust their appearance based on their parent window's appearance; it is not necessary to specify this attribute for a metal drawer.

You can read the state on Mac Classic, but you can only change it on Mac Carbon.

Returns false on any error.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.2.40 IsResizableMBS as Boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Whether this window is resizable.

Example:

```
Window1.IsResizableMBS=false
```

Notes: You can read the state on Mac Classic, but you can only change it on Mac Carbon.
Returns false on any error.

Use `GrowBoxTransparentMBS` on Composite Mac OS windows to enable the transparent grow box.

In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

Works in Cocoa.
(Read and Write computed property)

41.2.41 `IsZoomedMacMBS` as boolean

Plugin Version: 3.2, Platform: macOS, Targets: Desktop only.

Function: Returns true if the window is zoomed.

Example:

```
MsgBox str(window1.IsZoomedMacMBS)
```

Notes: Requires Mac OS 8.5 or newer.
Seems not to work correctly on RB 5.5.
In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

Added Cocoa support in plugin version 10.0.
(Read and Write computed property)

41.2.42 `IsZoomedMBS` as boolean

Platform: Windows, Targets: Desktop only.

Function: Returns true if this window has been maximized.

Example:

```
MsgBox str(window1.IsZoomedMBS)
```

Notes: If you set `IsZoomed` to true the window is maximized and if you set it to false the window size and position is restored.

In Xojo 2005 and newer you need to use `self.` in front of the method as the `propertyname` alone is not accepted.

Works on Cocoa.

(Read and Write computed property)

41.2.43 ModifiedMBS as boolean

Platform: macOS, Targets: Desktop only.

Function: You can set or get the value of the modified state.

Example:

```
mainwindow.modifiedMBS=true
```

Notes: Requires Mac OS 8.5 or newer.

As long as you don't set `modified` to false the window keeps to tell you that it's modified. Not sure why, so just set `modified` to false early in creating the window.

Added Cocoa support in plugin version 10.0.

(Read and Write computed property)

41.2.44 ToolbarVisibleMBS as boolean

Plugin Version: 4.0, Platform: macOS, Targets: Desktop only.

Function: Whether the toolbar is shown in this window or not.

Notes: Value is false on any error.

Only working on MacOS.

(Read and Write computed property)

41.2.45 TransparencyMBS as single

Plugin Version: 5.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The transparency of the window on Mac OS X, Windows 2000 and Windows XP.

Notes: 1 for opaque, 0 for invisible.

Return 1 on any error. On Windows it returns always 1 as the current transparency value can't be queried.

You need to call `MakeTransparent` before to install transparency.

(added Windows support in version 4.4)

Linux supported added with 14.0, but works only with Linux desktop which support alpha channel.

(Read and Write computed property)

41.2.46 UnifiedTitleAndToolbarMBS as Boolean

Plugin Version: 8.1, Platform: macOS, Targets: Desktop only.

Function: Whether this window has an unified title and toolbar look.

Example:

```
window1.UnifiedTitleAndToolbarMBS = true
```

Notes: This window draws its window title and toolbar using a unified appearance that has no separator between the two areas. A window may not have both UnifiedTitleAndToolbar and Metal appearance. If a window already has the metal attribute, attempting to set the Unified attribute will cause ChangeWindows to return an error, and vice versa. This constant was not added to this header file until Mac OS X 10.5, but it is actually available at runtime on Mac OS X 10.4 and later for windows of kDocumentWindowClass. However, on Mac OS X 10.5 and later, kHIWindowBitUnifiedTitleAndToolbar no longer has any effect, since all windows with toolbars now have a unified look.

(Read and Write computed property)

41.2.47 WindowProxyIconFileMBS as folderitem

Platform: macOS, Targets: Desktop only.

Function: That's the icon to the window which belongs to the file.

Example:

```
Var f as folderItem
```

```
f=getopenFolderItem("special/any")
if f<>nil then
window1.WindowProxyIconFileMBS=f
end if
```

Notes: Requires Mac OS 8.5 or newer.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

Added Cocoa support in plugin version 10.0.

(Read and Write computed property)

41.2.48 WinMenuHandleMBS as Integer

Platform: Windows, Targets: Desktop only.

Function: A property to access the handle used for the menu of a Window.

Example:

```
Var menu as Integer // global

if menu=0 then
menu=Window1.WinMenuHandleMBS // read it on the first window
else
Window1.winmenuHandleMBS=menu // set it on the second window
end if
```

Notes: Used in the example "Menu in every Window" to have a menubar in every window on Windows. But never forget to quit your app after all windows were closed.

In Xojo 2005 and newer you need to use self. in front of the method as the propertyname alone is not accepted.

(Read and Write computed property)

41.3 Globals

41.3.1 MenuBarHeightMBS as Integer

Platform: macOS, Targets: Desktop only.

Function: Returns the height of the menubar in pixels.

Example:

```
msgbox "The menubar is "+str(MenuBarHeightMBS)+" pixels height."
```

Notes: Requires the appearance manager.

Without plugin, you can use this:

```
declare function GetThemeMenuBarHeight lib "Carbon" (p as ptr) as Integer
```

```
Var l as Integer
```

```
Var p as memoryBlock
```

```
p=newmemoryBlock(10)
```

```
error=GetThemeMenuBarHeight(p) // 0 if okay.
```

```
menuheight=p.short(0) // in pixels. 22 on Mac OS X.
```

Added Cocoa support in plugin version 10.0.

Chapter 42

Windows

Chapter 43

XML

43.1 class XMLValidatorMBS

43.1.1 class XMLValidatorMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for the XML validator.

Notes: Using the validator in LibXML.

Blog Entries

- [News from the MBS Xojo Plugins Version 24.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.3](#)
- [MBS Xojo Plugins, version 24.3pr3](#)
- [News from the MBS Xojo Plugins Version 20.4](#)
- [MBS Xojo Plugins, version 20.4pr9](#)
- [MBS Xojo Plugins, version 20.2pr5](#)
- [MBS Xojo Plugins in version 19.2](#)
- [MBS Xojo Plugins, version 19.2pr8](#)
- [New XML Validator class for Xojo](#)

Xojo Developer Magazine

- [17.4, page 10: News](#)

43.1.2 Methods

43.1.3 CanonicalXML(XML as String, Mode as Integer = 0) as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates the canonical version of the XML.

Example:

```
Var x1 As String = "<test id='123'><f>1</f><f>2</f><f>3</f><f>4</f></test>"
Var x2 As String = XMLValidatorMBS.CanonicalXML(x1)
```

Break

Notes: Raises UnsupportedOperationException in case of a parse or serialization error.

The output mode:

Can be 0 for Original C14N 1.0 spec, 1 for Exclusive C14N 1.0 spec or 2 for C14N 1.1 spec.

43.1.4 Constructor(File as FolderItem)

Plugin Version: 20.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor taking file path to XML file.

Example:

```
Var f As FolderItem = SpecialFolder.Desktop.Child("schema.txt")
Var v As New XMLValidatorMBS(f)
```

Notes: May raise an XMLException if the XML can't be parsed or the validator can't be initialized. XML should have unix line endings (char 10), be saved as UTF-8 and itself a valid XML, so we can parse it correctly.

See also:

- 43.1.5 Constructor(XMLSchema as String)

1230

43.1.5 Constructor(XMLSchema as String)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor taking XML text.

Notes: May raise an XMLException if the XML can't be parsed or the validator can't be initialized.

For MBS Plugin 20.2 this function accepts also a file name/path to initialize.

XML should have unix line endings (char 10), be saved as UTF-8 and itself a valid XML, so we can parse it correctly.

See also:

- 43.1.4 Constructor(File as FolderItem)

1230

43.1.6 Destructor

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

43.1.7 LoadIconvLibrary(path as String, byref Error as String) as boolean

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads the iconv library.

Notes: The XML classes may use libiconv for text encoding conversion.

If you explicitly need, you can load the library on start of solution.

MBS Plugin may try to load iconv.dll/dylib/so automatically when first iconv function is called.

43.1.8 Message(index as Integer) as XMLValidatorMessageMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a message by index.

Notes: Index from 0 to MessageCount-1.

43.1.9 Messages as XMLValidatorMessageMBS()

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries array of messages recorded while validating.

43.1.10 SetCurrentWorkingDirectory(path as folderitem) as boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets current working directory for the application.

Notes: The library looks in current directory to find related files.

So if your schema or the XML file to check references other files, please set the directory.

Returns true on success or false on failure.

See also:

- 43.1.11 `SetCurrentWorkingDirectory(path as String)` as boolean 1232

43.1.11 `SetCurrentWorkingDirectory(path as String)` as boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets current working directory for the application.

Notes: The library looks in current directory to find related files.

So if your schema or the XML file to check references other files, please set the directory.

Returns true on success or false on failure.

See also:

- 43.1.10 `SetCurrentWorkingDirectory(path as folderitem)` as boolean 1231

43.1.12 `ValidateFile(file as FolderItem)` as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Validates XML file against a XML schema.

Example:

```
Var f As FolderItem = SpecialFolder.Desktop.Child("schema.txt")
Var v As New XMLValidatorMBS(f)
```

```
Var t As FolderItem = SpecialFolder.Desktop.Child("test.xml")
Var r As Integer = v.ValidateFile(t)
```

Notes: Validate a document tree in memory.

Returns 0 if the document is schemas valid, a positive error code number otherwise and -1 in case of internal or API error.

See messages array for the logged error and warning messages.

Please pass in XML file path and a valid XSD schema.

Use `SetCurrentWorkingDirectory` with `file.parent` to set working directory if needed.

XML should have unix line endings (char 10), be saved as UTF-8 and itself a valid XML, so we can parse it correctly.

See also:

- 43.1.13 ValidateFile(path as string) as Integer 1233

43.1.13 ValidateFile(path as string) as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Validates XML file against a XML schema.

Notes: Validate a document tree in memory.

Returns 0 if the document is schemas valid, a positive error code number otherwise and -1 in case of internal or API error.

See messages array for the logged error and warning messages.

Please pass in XML file path and a valid XSD schema.

Use SetCurrentWorkingDirectory with file.parent to set working directory if needed.

XML should have unix line endings (char 10), be saved as UTF-8 and itself a valid XML, so we can parse it correctly.

See also:

- 43.1.12 ValidateFile(file as FolderItem) as Integer 1232

43.1.14 ValidateString(text as string) as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Validates XML text against a XML schema.

Notes: Validate a document tree in memory.

Returns 0 if the document is schemas valid, a positive error code number otherwise and -1 in case of internal or API error.

See messages array for the logged error and warning messages.

XML should have unix line endings (char 10), be saved as UTF-8 and itself a valid XML, so we can parse it correctly.

43.1.15 Properties

43.1.16 MessageCount as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of messages we recorded in validation.

Notes: (Read only property)

43.1.17 Events

43.1.18 Error(message as XMLValidatorMessageMBS)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The error event called for validation errors.

43.1.19 Warning(message as XMLValidatorMessageMBS)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The warning event called for validation warnings.

43.2 class XMLValidatorMessageMBS

43.2.1 class XMLValidatorMessageMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for an error or warning message in XML validator.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [New XML Validator class for Xojo](#)

43.2.2 Methods

43.2.3 Constructor

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The private constructor.

43.2.4 Destructor

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor.

43.2.5 Properties

43.2.6 Code as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The error code.

Notes: (Read only property)

43.2.7 FileName as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The filename if available.

Notes: (Read only property)

43.2.8 IsError as Boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this was reported as error and not as warning.

Notes: (Read only property)

43.2.9 IsWarning as Boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this was reported as warning and not as error.

Notes: (Read only property)

43.2.10 Line as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The line number in the XML file if available.

Notes: (Read only property)

43.2.11 Message as String

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The message text.

Notes: (Read only property)

Chapter 44

XojoRuntime

44.1 Globals

44.1.1 AllObjectsOfClassMBS(ClassName as String) as Variant()

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Find all objects with given class using IsA operator.

Example:

```
Var AllWindows() As Variant = AllObjectsOfClassMBS("Window")
Var AllMenuItems() As Variant = AllObjectsOfClassMBS("MenuItem")
Var AllDictionary() As Variant = AllObjectsOfClassMBS("Dictionary")
```

[Break](#)

Notes: Similar to using Runtime.IterateObjects directly, loop over all objects and do a check with ISA operator on each one. The plugin is a bit more efficient doing this and returns matching items as array of variant.

Raises exception if class name is empty or not a known class to the Xojo runtime.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [MBS Xojo Plugins, version 21.1pr4](#)

44.1.2 ArrayDoubleMBS(paramArray values as Double) as Double()

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an array with given values.

Example:

```
// works
Var a() As Double = ArrayDoubleMBS(1, 2, 3)

// failes
Var b() As Double = Array(1,2,3)
```

Notes: With picking the right array function (ArrayDoubleMBS, ArrayStringMBS, ArrayInt64MBS, ArrayVariantMBS or ArrayIntegerMBS), you define what type of array you like. The Array function determinates the type based on value types, so it often isn't what we want.

Pass as many parameters as needed, just like Array() function.

Blog Entries

- [News from the MBS Xojo Plugins Version 20.5](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr3](#)

Xojo Developer Magazine

- [19.1, page 11: News](#)

44.1.3 ArrayInt64MBS(paramArray values as Int64) as Int64()

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an array with given values.

Example:

```
// works
Var a() As Int64 = ArrayInt64MBS(1, 2, 3.0)

// failes
Var b() As Int64 = Array(1,2,3.0)
```

Notes: With picking the right array function (ArrayDoubleMBS, ArrayStringMBS, ArrayInt64MBS, ArrayVariantMBS or ArrayIntegerMBS), you define what type of array you like. The Array function determinates the type based on value types, so it often isn't what we want.

Pass as many parameters as needed, just like Array() function.

Blog Entries

- [News from the MBS Xojo Plugins Version 20.5](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr3](#)

Xojo Developer Magazine

- [19.1, page 11: News](#)

44.1.4 ArrayIntegerMBS(paramArray values as Integer) as Integer()

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an array with given values.

Example:

```
// works
Var a() As Integer = ArrayIntegerMBS(1, 2, 3.0)

// failes
Var b() As Integer = Array(1,2,3.0)
```

Notes: With picking the right array function (ArrayDoubleMBS, ArrayStringMBS, ArrayInt64MBS, ArrayVariantMBS or ArrayIntegerMBS), you define what type of array you like. The Array function determinates the type based on value types, so it often isn't what we want.

Pass as many parameters as needed, just like Array() function.

Blog Entries

- [News from the MBS Xojo Plugins Version 20.5](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr3](#)

Xojo Developer Magazine

- [19.1, page 11: News](#)

44.1.5 ArrayIsAMBS(v as Variant, ClassName as string) as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if variant contains an array of the given class type.

Example:

```

Var w() as MouseCursor
w.Append System.Cursors.ArrowEastWest
w.Append System.Cursors.ArrowNorthSouth

```

```

Var v as Variant = w

```

```

if ArrayIsAMBS(w, "MouseCursor") then
MsgBox "OK"
end if

```

Notes: If the variant contains an array of some class, variant or object and has at least one value, the plugin checks this first value for being of type of the same class. Like an ISA check.

This helps to solve feedback case 12213.

Returns false if variant has no array, if variant is nil, if array is not an array of objects or if first object is nil.

Blog Entries

- [Useful array and variant functions](#)
- [MBS Real Studio Plugins, version 12.3pr11](#)

44.1.6 ArrayStringMBS(paramArray values as String) as String()

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an array with given values.

Example:

```

Var a() As String = ArrayStringMBS("Hello", "World")

```

Notes: With picking the right array function (ArrayDoubleMBS, ArrayStringMBS, ArrayInt64MBS, ArrayVariantMBS or ArrayIntegerMBS), you define what type of array you like. The Array function determinates the type based on value types, so it often isn't what we want.

Pass as many parameters as needed, just like Array() function.

Blog Entries

- [News from the MBS Xojo Plugins Version 20.5](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr3](#)

44.1.7 ArrayVariantMBS(paramArray values as Variant) as Variant()

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an array with given values.

Example:

```
Var a() As Variant = ArrayVariantMBS(1, 2, 3.0, True, False)
```

Notes: With picking the right array function (ArrayDoubleMBS, ArrayStringMBS, ArrayInt64MBS, ArrayVariantMBS or ArrayIntegerMBS), you define what type of array you like. The Array function determinates the type based on value types, so it often isn't what we want.

Pass as many parameters as needed, just like Array() function.

Blog Entries

- [News from the MBS Xojo Plugins Version 20.5](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr3](#)

Xojo Developer Magazine

- [19.1, page 11: News](#)

44.1.8 GetArrayAllocatedSizeMBS(v as variant) as integer

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Queries allocated array size in bytes.

Example:

```
Var a() As Integer // 8 byte per integer in 64-bit
```

```
System.DebugLog "allocated: "+Str(GetArrayAllocatedSizeMBS(a))
```

```
Redim a(9) // changes to 80 bytes
```

```
System.DebugLog "allocated: "+Str(GetArrayAllocatedSizeMBS(a))
```

```
Redim a(19) // changes to 160 bytes
```

```
System.DebugLog "allocated: "+Str(GetArrayAllocatedSizeMBS(a))
```

```
a.Append 123
```

```
System.DebugLog "allocated: "+Str(GetArrayAllocatedSizeMBS(a))
// shows 288, adds space for 16 more values
```

Notes: This function uses internal knowledge of the array layout in memory. So this will break in case a future Xojo version changes the layout. Please use only carefully for debugging.

Stopped working on Windows in recent Xojo versions, so we deprecated it.

44.1.9 GetArrayDataPointerMBS(v as variant) as Ptr

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries base pointer for the array data.

Notes: The pointer changes when append/insert/redim allocates new memory block.

This function uses internal knowledge of the array layout in memory. So this will break in case a future Xojo version changes the layout. Please use only carefully for debugging.

44.1.10 GetAutoMemoryAddressMBS(o as auto) as integer

Plugin Version: 17.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Queries memory address of auto object.

Example:

```
Var t As Auto = "Hello World"
```

```
Var o As Auto = "Hello "
```

```
o = o + "World"
```

```
Var s As Auto = t
```

```
Var ta As Integer = GetAutoMemoryAddressMBS(t)
```

```
Var oa As Integer = GetAutoMemoryAddressMBS(o)
```

```
Var sa As Integer = GetAutoMemoryAddressMBS(s)
```

```
// s and t show same address, but o is different
```

```
MsgBox Hex(ta)+EndOfLine+Hex(oa)+EndOfLine+Hex(sa)
```

```
// and s and t show address in const segment in the app, while o is dynamically allocated and has an heap address
```

Notes: Allows you to compare if two variables refer same object.

44.1.11 GetDelegateParametersMBS(del as variant) as String

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries parameter signature of a delegate.

Example:

```
Var aDelegate As ADelegate
aDelegate = AddressOf someFunction
```

```
Var p As String = GetDelegateParametersMBS(aDelegate)
MsgBox p
```

Notes: This function uses internal knowledge of the delegate layout in memory. So this will break in case a future Xojo version changes the layout. Please use only carefully for debugging.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [MBS Xojo Plugins, version 20.3pr8](#)
- [Inside Delegates](#)

44.1.12 GetDelegateTargetMBS(del as variant) as Variant

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries target object of a delegate.

Example:

```
Var aDelegate As ADelegate
aDelegate = AddressOf someFunction
```

```
Var target As Variant = GetDelegateTargetMBS(aDelegate)
MsgBox "target: "+Introspection.GetType(target).fullname
```

Notes: This function uses internal knowledge of the delegate layout in memory. So this will break in case a future Xojo version changes the layout. Please use only carefully for debugging.

See also Feedback case:

<http://feedback.xojo.com/case/23305>

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [MBS Xojo Plugins, version 20.3pr8](#)
- [Inside Delegates](#)

44.1.13 GetDelegateWeakMBS(del as variant) as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries weak flag of a delegate.

Example:

```
Var aDelegate As ADelegate
aDelegate = AddressOf someFunction
```

```
Var weak As Boolean = GetDelegateWeakMBS(aDelegate)
MsgBox "weak: "+Str(weak)
```

Notes: This function uses internal knowledge of the delegate layout in memory. So this will break in case a future Xojo version changes the layout. Please use only carefully for debugging.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [MBS Xojo Plugins, version 20.3pr8](#)
- [Inside Delegates](#)

44.1.14 GetObjectMemoryAddressMBS(o as object) as integer

Plugin Version: 17.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries memory address of object.

Example:

```
Var a As Integer = GetObjectMemoryAddressMBS(app)
```

```
// show address
MsgBox Hex(a)
```

Notes: Allows you to compare if two variables refer same object.

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr8](#)

44.1.15 GetObjectReferenceCountMBS(o as object) as integer

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries object reference count.

Example:

```
Var o As Object = window1
```

```
System.DebugLog "Reference count: "+Str(GetObjectReferenceCountMBS(o))
```

```
Var k as Object = o
```

```
System.DebugLog "Reference count: "+Str(GetObjectReferenceCountMBS(o))
```

```
Var v as Variant = o
```

```
System.DebugLog "Reference count: "+Str(GetObjectReferenceCountMBS(o))
```

```
v = Nil
```

```
k = nil
```

```
System.DebugLog "Reference count: "+Str(GetObjectReferenceCountMBS(o))
```

Notes: This function uses internal knowledge of the object layout in memory. So this will break in case a future Xojo version changes the layout. Please use only carefully for debugging.

44.1.16 GetStringMemoryAddressMBS(s as string) as integer

Plugin Version: 17.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries memory address of string.

Example:

```
Var t As String = "Hello World"
```

```
Var o As String = "Hello "
```

```
o = o + "World"
```

```
Var s As String = t
```

```
Var ta As Integer = GetStringMemoryAddressMBS(t)
```

```
Var oa As Integer = GetStringMemoryAddressMBS(o)
```

```
Var sa As Integer = GetStringMemoryAddressMBS(s)
```

```
// s and t show same address, but o is different
```

```
MsgBox Hex(ta)+EndOfLine+Hex(oa)+EndOfLine+Hex(sa)
```

// and s and t show address in const segment in the app, while o is dynamically allocated and has an heap address

Notes: Allows you to compare if two variables refer same string.

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr8](#)

44.1.17 GetStringReferenceCountMBS(s as string) as integer

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries string reference count.

Example:

```
Var a As String = "World"
```

```
Var s As String = "Hello "+a // use + to make new string
```

```
System.DebugLog "Reference count: "+Str(GetStringReferenceCountMBS(s))
```

```
Var k As String = s
```

```
System.DebugLog "Reference count: "+Str(GetStringReferenceCountMBS(s))
```

```
Var v As Variant = s
```

```
System.DebugLog "Reference count: "+Str(GetStringReferenceCountMBS(s))
```

```
v = ""
```

```
k = ""
```

```
System.DebugLog "Reference count: "+Str(GetStringReferenceCountMBS(s))
```

Notes: This function uses internal knowledge of the string layout in memory.

So this will break in case a future Xojo version changes the layout.

Please use only carefully for debugging.

44.1.18 GetTextMemoryAddressMBS(s as text) as integer

Plugin Version: 17.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Queries memory address of text object.

Example:

```
Var t As Text = "Hello World"  
Var n As Integer = GetTextMemoryAddressMBS(t)  
MsgBox Hex(n)
```

Notes: Allows you to compare if two variables refer same text.

44.1.19 GetVariantArrayMBS(VariantContainingArray as Variant) as Variant()

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a variant containing array for an array and returns it as an array of variant.

Example:

```
Var test() as Dictionary  
test.Append new Dictionary  
  
// this fails  
Var vv As Variant = test  
Var t() As Variant = vv  
  
// this works!  
Var v() as Variant = GetVariantArrayMBS(test)  
Var dic as Dictionary = v(0)  
Break
```

Notes: Works with all arrays of objects (any type).

Raises exception if the array can't be converted or the variant contains no array.

Blog Entries

- [MBS Xojo Plugins, version 24.3pr6](#)
- [Useful array and variant functions](#)
- [Variant Helper functions in MBS Xojo Plugins](#)
- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.0](#)
- [MBS Xojo / Real Studio Plugins, version 14.0pr4](#)

Xojo Developer Magazine

- [12.2, page 10: News](#)

44.1.20 GetVariantArrayUboundMBS(v as Variant) as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries array ubound for an array inside a variant.

Notes: If you have an array in the variant, you'd normally assign it to an array to query values from array. This function queries ubound directly.

Returns -3 if variant is nil and -2 if variant contains not an object array.

Updated in 13.5 plugins to raise exception is array is nil.

Blog Entries

- [Useful array and variant functions](#)
- [Variant Helper functions in MBS Xojo Plugins](#)
- [MBS Xojo / Real Studio Plugins, version 13.5pr8](#)
- [MBS Real Studio Plugins, version 13.0pr8](#)

44.1.21 GetVariantArrayValueMBS(v as Variant, index as Integer) as Variant

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a variant from an array inside a variant.

Example:

```
// some part of app builds an array
Var a() as pair
a.Append 1:"Hello"
a.Append 2:"World"

// passes it as Variant somewhere else

Var v as Variant = a

// and later you may want to get values back without knowing the array type

// this raises TypeMismatchException
'dim o() as Object = v

// so use plugin to get objects:
Var v1 as Variant = GetVariantArrayValueMBS(v, 0)
Var v2 as Variant = GetVariantArrayValueMBS(v, 1)

// now you can check type and cast to the object type
Var p1 as pair = v1
Var p2 as pair = v2
```



```
MsgBox p1.Right+" "+p2.Right
```

Notes: This function is to allow getting objects from an array inside a variant without known the class used to declare array.

Returns nil on any error.

Updated in 13.5 plugins to raise exception is array is nil.

Also updated in v13.5 to work with object, variant, string, date, integer, double, single, boolean, Int64 arrays. Other array types will raise exception.

Blog Entries

- [MBS Xojo Plugins, version 24.3pr6](#)
- [Useful array and variant functions](#)
- [Variant Helper functions in MBS Xojo Plugins](#)
- [MBS Xojo / Real Studio Plugins, version 13.5pr8](#)
- [MBS Real Studio Plugins, version 13.0pr8](#)

44.1.22 GetVariantAsDictionaryArrayMBS(v as variant) as Dictionary()

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Casts a variant to a dictionary array.

Example:

```
Var d As New Dictionary
Var v() As Variant
v.Append d
v.Append Nil

Var va as Variant = v
'Dim dic() As Dictionary = va // type mismatch exception
Var dic() As Dictionary = GetVariantAsDictionaryArrayMBS(va) // works
```

Break

Notes: Raises exception if variant does not have an object/variant array. If it contains a non-dictionary object, we raise an exception.

Blog Entries

- [Useful array and variant functions](#)
- [Variant Helper functions in MBS Xojo Plugins](#)

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 20.4](#)
- [MBS Xojo Plugins, version 20.4pr7](#)

44.1.23 GetVariantTypeMBS(va as variant) as Integer

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries variant type.

Example:

```
Var i As Int32 = 5
Var u As UInt32 = 5
Var vi As Variant = i
Var vu As Variant = u
Var ti As Integer = vi.Type // 2
Var tu As Integer = vu.Type // 2
Var mi As Integer = GetVariantTypeMBS(vi) // 2
Var mu As Integer = GetVariantTypeMBS(vu) // 102
```

`Break // check in debugger`

Notes: Same as `variant.type` or `VarType()` function, but returns 102 for `UInt32` and 103 for `UInt64`. This way plugin functions internally can distinguish between variants with unsigned vs. signed data types.

Blog Entries

- [Useful array and variant functions](#)
- [News from the MBS Xojo Plugins Version 20.5](#)
- [The VariantTypeString function for Xojo](#)
- [MBS Xojo Plugins in version 20.5](#)
- [MBS Xojo Plugins, version 20.5pr2](#)
- [Variant Helper functions in MBS Xojo Plugins](#)

Xojo Developer Magazine

- [19.1, page 11: News](#)

44.1.24 ObjectIsAMBS(o as object, ClassName as string) as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if a given object is of a given class.

Example:

```
// you must know exact name of class

Var v as Variant = window1

if ObjectIsAMBS(v, "window1.window1") then
MsgBox "OK"
end if

v = System.Cursors.MagnifyLarger

if ObjectIsAMBS(v, "MouseCursor") then
MsgBox "OK"
end if
```

Notes: Like ISA operator, but with class name as string.

Returns false if o is nil.

Blog Entries

- [Useful array and variant functions](#)
- [MBS Real Studio Plugins, version 12.3pr11](#)

44.1.25 SetVariantArrayValueMBS(v as Variant, index as Integer, value as Variant)

Plugin Version: 16.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an object from an array of objects inside a variant.

Example:

```
// some part of app builds an array
Var a() as pair

// passes it as Variant somewhere else

Var v as Variant = a

// and later you may want to put values in array without knowing object type for array

// this raises TypeMismatchException
'dim o() as Object = v

// so use plugin

SetVariantArrayValueMBS(v, 0, 1:"Hello")
SetVariantArrayValueMBS(v, 1, 2:"World")
```

```
MsgBox a(0).Right+" "+a(1).Right
```

Notes: This function is to allow setting objects in an array inside a variant without known the class used to declare array.

Also updated to work with object, variant, string, date, integer, double, single, boolean, Int64 arrays. Other array types will raise exception.

Be sure to only put objects of right class in the array! Else you risk crashes.

If index is 1 bigger than ubound, we append an element.

Blog Entries

- [MBS Xojo Plugins, version 24.3pr6](#)
- [Useful array and variant functions](#)
- [Variant Helper functions in MBS Xojo Plugins](#)
- [MBS Xojo / Real Studio Plugins, version 16.2pr3](#)

44.1.26 GetEncodingOfStringMBS(s as string) as UInt32

Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the internal value for the encoding.

Example:

```
Var s as string = "Hello"
```

```
MsgBox hex(GetEncodingOfStringMBS(s)) // shows 8000100 for UTF8
```

Notes: Only useful on Xojo 4.5 and newer.

Some example values for encoding:

MacRoman	0	Also for ASCII or binary data used.
WindowsLatin1	&h0500	ANSI codepage 1252
ISOLatin1	&h0201	ISO 8859-1
NextStepLatin	&h0B01	NextStep encoding
Unicode	&h0100	16 bit Unicode
UTF8	&h0800100	8 bit Unicode
Invalid	&hFFFFFFFF	(Binary)
Invalid	&hFFFF	(Binary)

Renamed from GetStringEncoding to GetEncodingOfString in MBS Plugin 3.1.

Xojo Developer Magazine

- [2.4, page 43: Cross platform streams, We write our own binary stream to save our data by Christian Schmitz](#)

44.1.27 SetEncodingOfStringMBS(s as string, encoding as UInt32)

Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the internal value for the encoding of this string.

Example:

```
Var s as string = "√§√√°"
Var t as string = ConvertEncoding(s, encodings.UTF16)
Var m as MemoryBlock = t // memoryblock has bytes from UTF16 string without knowing the encoding
Var u as string = m // convert back to a string without encoding
```

```
MsgBox u // shows wrong characters
SetEncodingOfStringMBS u, 256 // set to UTF-16
MsgBox u
```

Notes: Only useful on Xojo 4.5 and newer.

Some example values for encoding:

MacRoman	0	Also for ASCII or binary data used.
WindowsLatin1	&h0500	ANSI codepage 1252
ISOLatin1	&h0201	ISO 8859-1
NextStepLatin	&h0B01	NextStep encoding
Unicode	&h0100	16 bit Unicode
UTF8	&h08000100	8 bit Unicode
Invalid	&hFFFFFFFF	(Binary)
Invalid	&hFFFF	(Binary)

Renamed from SetStringEncoding to SetEncodingOfString in MBS Plugin 3.1.

Xojo Developer Magazine

- [2.4, page 43: Cross platform streams, We write our own binary stream to save our data](#) by Christian Schmitz

Chapter 45

List of Questions in the FAQ

- 46.0.1 Can anyone help me convert seconds to time in this format hh:mm:ss? 1265
- 46.0.2 Do you have plugins for Android? 1266
- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1266
- 46.0.4 How to catch delete key? 1267
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.6 How to delete a folder? 1269
- 46.0.7 How to detect if CPU is 64bit processor? 1270
- 46.0.8 How to query variant type string for a variant? 1271
- 46.0.9 How to refresh a htmlviewer on Windows? 1272
- 46.0.10 Is there an example for vector graphics in Xojo? 1273
- 46.0.11 Picture functions do not preserve resolution values? 1274
- 46.0.12 A toolbox call needs a rect - how do I give it one? 1274
- 46.0.13 API client not supported? 1274
- 46.0.14 Can I access Access Database with Java classes? 1275
- 46.0.15 Can I create PDF from Xojo Report using DynaPDF? 1276
- 46.0.16 Can I use AppleScripts in a web application? 1276
- 46.0.17 Can I use graphics class with DynaPDF? 1276
- 46.0.18 Can I use sockets on a web application? 1277
- 46.0.19 Can I use your ChartDirector plugin on a web application? 1277

- 46.0.20 Can I use your DynaPDF plugin on a web application? 1278
- 46.0.21 Can I use your plugin controls on a web application? 1279
- 46.0.22 Can you get an unique machine ID? 1279
- 46.0.23 ChartDirector: Alignment Specification 1279
- 46.0.24 ChartDirector: Color Specification 1280
- 46.0.25 ChartDirector: Font Specification 1284
- 46.0.26 ChartDirector: Mark Up Language 1287
- 46.0.27 ChartDirector: Parameter Substitution and Formatting 1291
- 46.0.28 ChartDirector: Shape Specification 1295
- 46.0.29 Copy styled text? 1296
- 46.0.30 Do you have code to validate a credit card number? 1297
- 46.0.31 Do you have plugins for X-Rite EyeOne, eXact or i1Pro? 1297
- 46.0.32 Does SQL Plugin handle stored procedures with multiple result sets? 1298
- 46.0.33 Does the plugin home home? 1298
- 46.0.34 folderitem.absolutePath is limited to 255 chars. How can I get longer ones? 1298
- 46.0.35 Has anyone played round with using CoreImage to do things like add dissolve transitions say when changing from one tab to another within a window? 1299
- 46.0.36 How about Plugin support for older OS X? 1300
- 46.0.37 How can I detect whether an Intel CPU is a 64bit CPU? 1300
- 46.0.38 How can I disable the close box of a window on Windows? 1301
- 46.0.39 How can I get all the environment variables from Windows? 1302
- 46.0.40 How can i get similar behavior to Roxio Toast or iTunes where clicking a 'burn' button allows the next inserted blank CD-R to bypass the Finder and be accepted by my application? 1303
- 46.0.41 How can I get text from a PDF? 1303
- 46.0.42 How can I get text from a Word Document? 1303
- 46.0.43 How can I get the item string for a given file creator? 1303
- 46.0.44 How can I launch an app using it's creator code? 1305
- 46.0.45 How can I learn what shared libraries are required by a plugin on Linux? 1305
- 46.0.46 How can I validate an email address? 1306
- 46.0.47 How do I decode correctly an email subject? 1307

	1257
• 46.0.48 How do I enable/disable a single tab in a tabpanel?	1308
• 46.0.49 How do I find the root volume for a file?	1308
• 46.0.50 How do I get the current languages list?	1309
• 46.0.51 How do I get the Mac OS Version?	1310
• 46.0.52 How do I get the printer name?	1311
• 46.0.53 How do I make a metal window if RB does not allow me this?	1311
• 46.0.54 How do I make a smooth color transition?	1312
• 46.0.55 How do I read the applications in the dock app?	1313
• 46.0.56 How do I truncate a file?	1314
• 46.0.57 How do update a Finder's windows after changing some files?	1314
• 46.0.58 How to access a USB device directly?	1314
• 46.0.59 How to add icon to file on Mac?	1315
• 46.0.60 How to ask the Mac for the Name of the Machine?	1315
• 46.0.61 How to automatically enable retina in my apps?	1315
• 46.0.62 How to avoid leaks with Cocoa functions?	1316
• 46.0.63 How to avoid trouble connecting to oracle database with SQL Plugin?	1316
• 46.0.64 How to avoid ___NSAutoreleaseNoPool console messages in threads?	1317
• 46.0.65 How to bring app to front?	1317
• 46.0.66 How to bring my application to front?	1318
• 46.0.67 How to catch Control-C on Mac or Linux in a console app?	1318
• 46.0.68 How to change name of application menu?	1319
• 46.0.69 How to change the name in the menubar of my app on Mac OS X?	1319
• 46.0.70 How to check if a folder/directory has subfolders?	1320
• 46.0.71 How to check if Macbook runs on battery or AC power?	1320
• 46.0.72 How to check if Microsoft Outlook is installed?	1321
• 46.0.73 How to check on Mac OS which country or language is currently selected?	1322
• 46.0.74 How to code sign my app with plugins?	1322
• 46.0.75 How to collapse a window?	1323
• 46.0.76 How to compare two pictures?	1323

- 46.0.77 How to compile PHP library? 1325
- 46.0.78 How to convert a `BrowserType` to a `String` with `WebSession.Browser`? 1326
- 46.0.79 How to convert a `EngineType` to a `String` with `WebSession.Engine`? 1327
- 46.0.80 How to convert a `PlatformType` to a `String` with `WebSession.Platform`? 1327
- 46.0.81 How to convert a text to iso-8859-1 using the `TextEncoder`? 1328
- 46.0.82 How to convert `ChartTime` back to Xojo date? 1329
- 46.0.83 How to convert line endings in text files? 1329
- 46.0.84 How to convert picture to string and back? 1330
- 46.0.85 How to copy an array? 1331
- 46.0.86 How to copy an dictionary? 1331
- 46.0.87 How to copy parts of a movie to another one? 1332
- 46.0.88 How to create a birthday like calendar event? 1332
- 46.0.89 How to create a GUID? 1333
- 46.0.90 How to create a Mac picture clip file? 1333
- 46.0.91 How to create a PDF file in Xojo? 1334
- 46.0.92 How to create `EmailAttachment` for PDF Data in memory? 1334
- 46.0.93 How to create PDF for image files? 1335
- 46.0.94 How to CURL Options translate to Plugin Calls? 1336
- 46.0.95 How to delete file with ftp and curl plugin? 1337
- 46.0.96 How to detect display resolution changed? 1337
- 46.0.97 How to detect retina? 1338
- 46.0.98 How to disable force quit? 1338
- 46.0.99 How to disable the error dialogs from Internet Explorer on javascript errors? 1338
- 46.0.100 How to display a PDF file in Xojo? 1338
- 46.0.101 How to do a lottery in RB? 1339
- 46.0.102 How to do an asycron DNS lookup? 1340
- 46.0.103 How to draw a dushed pattern line? 1340
- 46.0.104 How to draw a nice antialiased line? 1341
- 46.0.105 How to dump java class interface? 1342

	1259
• 46.0.106 How to duplicate a picture with mask or alpha channel?	1342
• 46.0.107 How to enable assistive devices?	1343
• 46.0.108 How to encrypt a file with Blowfish?	1344
• 46.0.109 How to extract text from HTML?	1344
• 46.0.110 How to find empty folders in a folder?	1345
• 46.0.111 How to find iTunes on a Mac OS X machine fast?	1345
• 46.0.112 How to find network interface for a socket by it's name?	1346
• 46.0.113 How to find version of Microsoft Word?	1347
• 46.0.114 How to fix CURL error 60/53 on connecting to server?	1347
• 46.0.115 How to format double with n digits?	1348
• 46.0.116 How to get a time converted to user time zone in a web app?	1348
• 46.0.117 How to get an handle to the frontmost window on Windows?	1349
• 46.0.118 How to get CFAbsoluteTime from date?	1349
• 46.0.119 How to get client IP address on web app?	1350
• 46.0.120 How to get fonts to load in charts on Linux?	1350
• 46.0.121 How to get fonts to load in DynaPDF on Linux?	1351
• 46.0.122 How to get GMT time and back?	1351
• 46.0.123 How to get good crash reports?	1352
• 46.0.124 How to get list of all threads?	1352
• 46.0.125 How to get parameters from webpage URL in Xojo Web Edition?	1353
• 46.0.126 How to get the color for disabled textcolor?	1353
• 46.0.127 How to get the current free stack space?	1354
• 46.0.128 How to get the current timezone?	1354
• 46.0.129 How to get the current window title?	1356
• 46.0.130 How to get the cursor blink interval time?	1357
• 46.0.131 How to get the list of the current selected files in the Finder?	1357
• 46.0.132 How to get the Mac OS system version?	1358
• 46.0.133 How to get the Mac OS Version using System.Gestalt?	1358
• 46.0.134 How to get the screensize excluding the task bar?	1359

- 46.0.135 How to get the size of the frontmost window on Windows? 1359
- 46.0.136 How to get the source code of a HTMLViewer? 1360
- 46.0.137 How to get Xojo apps running Linux? 1360
- 46.0.138 How to handle really huge images with GraphicsMagick or ImageMagick? 1361
- 46.0.139 How to handle tab key for editable cells in listbox? 1361
- 46.0.140 How to hard link MapKit framework? 1362
- 46.0.141 How to have a PDF downloaded to the user in a web application? 1363
- 46.0.142 How to hide all applications except mine? 1363
- 46.0.143 How to hide script errors in HTMLViewer on Windows? 1364
- 46.0.144 How to hide the grid/background/border in ChartDirector? 1364
- 46.0.145 How to hide the mouse cursor on Mac? 1364
- 46.0.146 How to insert image to NSTextView or TextArea? 1364
- 46.0.147 How to jump to an anchor in a htmlviewer? 1365
- 46.0.148 How to keep a movieplayer unclickable? 1366
- 46.0.149 How to keep my web app from using 100% CPU time? 1366
- 46.0.150 How to kill a process by name? 1366
- 46.0.151 How to know how many CPUs are present? 1367
- 46.0.152 How to know the calling function? 1367
- 46.0.153 How to launch an app using it's creator code? 1368
- 46.0.154 How to launch disc utility? 1368
- 46.0.155 How to make a lot of changes to a REAL SQL Database faster? 1369
- 46.0.156 How to make a NSImage object for my retina enabled app? 1369
- 46.0.157 How to make a window borderless on Windows? 1370
- 46.0.158 How to make an alias using AppleEvents? 1370
- 46.0.159 How to make AppleScripts much faster? 1371
- 46.0.160 How to make double clicks on a canvas? 1371
- 46.0.161 How to make my Mac not sleeping? 1373
- 46.0.162 How to make my own registration code scheme? 1374
- 46.0.163 How to make small controls on Mac OS X? 1374

	1261
• 46.0.164 How to mark my Mac app as background only?	1375
• 46.0.165 How to move a file or folder to trash?	1375
• 46.0.166 How to move an application to the front using the creator code?	1376
• 46.0.167 How to move file with ftp and curl plugin?	1377
• 46.0.168 How to normalize string on Mac?	1377
• 46.0.169 How to obscure the mouse cursor on Mac?	1378
• 46.0.170 How to open icon file on Mac?	1378
• 46.0.171 How to open PDF in acrobat reader?	1378
• 46.0.172 How to open printer preferences on Mac?	1379
• 46.0.173 How to open special characters panel on Mac?	1379
• 46.0.174 How to optimize picture loading in Web Edition?	1380
• 46.0.175 How to parse XML?	1380
• 46.0.176 How to play audio in a web app?	1381
• 46.0.177 How to pretty print xml?	1382
• 46.0.178 How to print to PDF?	1382
• 46.0.179 How to query Spotlight's Last Open Date for a file?	1383
• 46.0.180 How to quit windows?	1383
• 46.0.181 How to read a CSV file correctly?	1384
• 46.0.182 How to read the command line on windows?	1384
• 46.0.183 How to render PDF pages with PDF Kit?	1385
• 46.0.184 How to restart a Mac?	1386
• 46.0.185 How to resume ftp upload with curl plugin?	1386
• 46.0.186 How to rotate a PDF page with CoreGraphics?	1386
• 46.0.187 How to rotate image with CoreImage?	1387
• 46.0.188 How to run a 32 bit application on a 64 bit Linux?	1388
• 46.0.189 How to save HTMLViewer to PDF with landscape orientation?	1388
• 46.0.190 How to save RTFD?	1389
• 46.0.191 How to save RTFD?	1389
• 46.0.192 How to scale a picture proportionally with mask?	1390

- 46.0.193 How to scale a picture proportionally? 1391
- 46.0.194 How to scale/resize a CIImageMBS? 1392
- 46.0.195 How to scale/resize a picture? 1392
- 46.0.196 How to search with regex and use unicode codepoints? 1393
- 46.0.197 How to see if a file is invisible for Mac OS X? 1394
- 46.0.198 How to set cache size for SQLite or REALSQLDatabase? 1394
- 46.0.199 How to set the modified dot in the window? 1395
- 46.0.200 How to show a PDF file to the user in a Web Application? 1395
- 46.0.201 How to show Keyboard Viewer programmatically? 1395
- 46.0.202 How to show the mouse cursor on Mac? 1396
- 46.0.203 How to shutdown a Mac? 1397
- 46.0.204 How to sleep a Mac? 1397
- 46.0.205 How to speed up rasterizer for displaying PDFs with DynaPDF? 1397
- 46.0.206 How to use PDFLib in my RB application? 1398
- 46.0.207 How to use quotes in a string? 1398
- 46.0.208 How to use Sybase in Web App? 1398
- 46.0.209 How to use the Application Support folder? 1399
- 46.0.210 How to use the IOPMCopyScheduledPowerEvents function in Xojo? 1399
- 46.0.211 How to validate a GUID? 1402
- 46.0.212 How to walk a folder hierarchie non recursively? 1402
- 46.0.213 I got this error: PropVal, QDPictMBS.Name (property value), Type mismatch error. Expected CGDataProviderMBS, but got Variant, Name:QDPictMBS 1403
- 46.0.214 I registered the MBS Plugins in my application, but later the registration dialog is shown. 1403
- 46.0.215 I want to accept Drag & Drop from iTunes 1404
- 46.0.216 I'm drawing into a listbox but don't see something. 1406
- 46.0.217 I'm searching for a method or so to move a window from position x.y to somewhere else on the screen. 1406
- 46.0.218 If I use one of your plug-ins under windows, would this then impose the use of dll after compilation or my would my compiled soft still be a stand-alone single file software? 1407
- 46.0.219 Is the fn key on a powerbook keyboard down? 1407

	1263
• 46.0.220 Is there a case sensitive Dictionary?	1407
• 46.0.221 Is there a way to use the MBS plugin to get only the visible item and folder count on a volume?	1408
• 46.0.222 Is there an easy way I can launch the Displays preferences panel?	1408
• 46.0.223 List of Windows Error codes?	1409
• 46.0.224 Midi latency on Windows problem?	1409
• 46.0.225 My Xojo Web App does not launch. Why?	1409
• 46.0.226 SQLiteDatabase not initialized error?	1410
• 46.0.227 Textconverter returns only the first x characters. Why?	1410
• 46.0.228 The type translation between CoreFoundation/Foundation and Xojo data types.	1411
• 46.0.229 Uploaded my web app with FTP, but it does not run on the server!	1413
• 46.0.230 What classes to use for hotkeys?	1413
• 46.0.231 What do I need for Linux to get picture functions working?	1414
• 46.0.232 What does the NAN code mean?	1414
• 46.0.233 What font is used as a 'small font' in typical Mac OS X apps?	1415
• 46.0.234 What is last plugin version to run on Mac OS X 10.4?	1415
• 46.0.235 What is last plugin version to run on PPC?	1416
• 46.0.236 What is last version of the plugins for macOS 32-bit?	1416
• 46.0.237 What is the difference between Timer and WebTimer?	1416
• 46.0.238 What is the list of Excel functions?	1416
• 46.0.239 What is the replacement for PluginMBS?	1417
• 46.0.240 What to do on Xojo reporting a conflict?	1418
• 46.0.241 What to do with a NSImageCacheException?	1418
• 46.0.242 What to do with MySQL Error 2014?	1418
• 46.0.243 What to do with SQL Plugin reporting Malformed string as error?	1418
• 46.0.244 Where is CGGetActiveDisplayListMBS?	1419
• 46.0.245 Where is CGGetDisplaysWithPointMBS?	1419
• 46.0.246 Where is CGGetDisplaysWithRectMBS?	1419
• 46.0.247 Where is CGGetOnlineDisplayListMBS?	1419
• 46.0.248 Where is GetObjectClassNameMBS?	1419

- 46.0.249 Where is NetworkAvailableMBS? 1420
- 46.0.250 Where is StringHeight function in DynaPDF? 1420
- 46.0.251 Where is XLSDocumentMBS class? 1420
- 46.0.252 Where to get information about file formats? 1421
- 46.0.253 Where to register creator code for my application? 1421
- 46.0.254 Which Mac OS X frameworks are 64bit only? 1421
- 46.0.255 Which plugins are 64bit only? 1422
- 46.0.256 Why application doesn't launch because of a missing ddraw.dll!? 1422
- 46.0.257 Why application doesn't launch because of a missing shlwapi.dll!? 1422
- 46.0.258 Why do I hear a beep on keydown? 1422
- 46.0.259 Why does folderitem.item return nil? 1422
- 46.0.260 Why doesn't showurl work? 1423
- 46.0.261 Why don't the picture functions not work on Linux? 1423
- 46.0.262 Why have I no values in my chart? 1423
- 46.0.263 Will application size increase with using plugins? 1423
- 46.0.264 XLS: Custom format string guidelines 1424
- 46.0.265 Xojo doesn't work with your plugins on Windows 98. 1425
- 46.0.266 Xojo or my RB application itself crashes on launch on Mac OS Classic. Why? 1425

Chapter 46

The FAQ

46.0.1 Can anyone help me convert seconds to time in this format hh:mm:ss?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sure, here's a routine I use (which has an advantage over the previously-posted Date-based solution in that you don't have to rely on the creation of an object – all that happens is some division and string concatenation):

Example:

```
Function SecsToTimeString(timeInSecs as Integer, padHours as boolean, padMinutes as boolean) as string
// Given an amount time (in seconds), generates a string representing that amount
// of time. The padHours and padMinutes parameters determine whether to display
// hours and minutes if their values are zero.
```

```
// Examples:
// timeInSecs = 90, padHours = true; returns "00:01:30"
// timeInSecs = 1, padHours = false, padMinutes = true; returns "00:01"
// timeInSecs = 3601, padMinutes = false; returns "01:00:01"
```

```
Var hours, minutes, seconds as Integer
Var hoursString, minutesString as string
```

```
hours = timeInSecs / 3600
minutes = (timeInSecs mod 3600) / 60
seconds = timeInSecs mod 60
```

```
if hours = 0 then
if padHours then
hoursString = "00:"
else
hoursString = ""
end if
```

```

else
hoursString = Format(hours, "##\:")
end if
if minutes = 0 then
if hours <>0 or padMinutes then
minutesString = "00:"
else
minutesString = ""
end if
else
minutesString = Format(minutes, "00\:")
end if

return hoursString + minutesString + Format(seconds, "00")
End Function

```

Notes: (from the rb mailinglist)

46.0.2 Do you have plugins for Android?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Since there is no plugin SDK for Android, we have no way to make a plugin for Android.

Notes: We support macOS, Windows, Linux and iOS.

46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use functions from NSColor to get proper highlight color in RGB:

Example:

```

Function ProperHighlightColor(active as Boolean) As Color
#if TargetCocoa
Var theColor As NSColorMBS
If active Then
theColor = NSColorMBS.alternateSelectedControlColor
Else
theColor = NSColorMBS.secondarySelectedControlColor
End If

```

```

Var rgbColor As NSColorMBS = theColor.colorUsingColorSpaceName(NSColorSpaceMBS.NSCalibratedRGB-
BColorSpace)

```

```

If rgbColor <>Nil Then
Var red as Integer = rgbColor.redComponent * 255.0
Var green as Integer = rgbColor.greenComponent * 255.0
Var blue as Integer = rgbColor.blueComponent * 255.0
Return RGB(red, green, blue)
Else
Return HighlightColor
End If
#else
return HighlightColor
#endif
End Function

```

Notes: As you see we convert color to Calibrated RGB for best results.
See also:

- 46.0.4 How to catch delete key? 1267
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.6 How to delete a folder? 1269
- 46.0.7 How to detect if CPU if 64bit processor? 1270
- 46.0.8 How to query variant type string for a variant? 1271
- 46.0.9 How to refresh a htmlviewer on Windows? 1272

46.0.4 How to catch delete key?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following is the code in keydown event catches delete or backspace keys.

Example:

```

Function KeyDown(Key As String) As Boolean
if asc(key) = 8 or asc(key) = 127 then
MsgBox "Delete"
Return true
end if
End Function

```

See also:

- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1266
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.6 How to delete a folder? 1269

- 46.0.7 How to detect if CPU is 64bit processor? 1270
- 46.0.8 How to query variant type string for a variant? 1271
- 46.0.9 How to refresh a htmlviewer on Windows? 1272

46.0.5 How to convert cmyk to rgb?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

The following is the code to convert cmyk values to an RGB color datatype.

It's just a basic estimate of the color values. If you are looking for completely color accurate solution, this is not it. It should work for most people. :)

Example:

```
Function CMYKToRGB(c as Integer, m as Integer, y as Integer, k as Integer) As color
// converts c,m,y,k values (0-100) to color data type RGB
// place this in a method. Supply C,M,Y,K values-
// it returns color datatype
```

```
Var color_RGB as color
Var r, g, b as Integer

r=255-round(2.55*(c+k))
if r<0 then
r=0
end if
g=255-round(2.55*(m+k))
if g<0 then
g=0
end if
b=255-round(2.55*(y+k))
if b<0 then
b=0
end if

color_RGB=RGB(r,g,b)

return color_RGB

End Function
```

Notes:

(from the rb mailinglist)

See also:

- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1269
- 46.0.4 How to catch delete key? 1267
- 46.0.6 How to delete a folder? 1269
- 46.0.7 How to detect if CPU is 64bit processor? 1270
- 46.0.8 How to query variant type string for a variant? 1271
- 46.0.9 How to refresh a htmlviewer on Windows? 1272

46.0.6 How to delete a folder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following is the code that deletes a folder recursively.

Example:

```
Sub deletefolder(f as folderitem)
```

```
Var files(-1) as FolderItem
```

```
if f=nil then Return
```

```
// delete single file
```

```
if f.Directory=false then
```

```
f.Delete
```

```
Return
```

```
end if
```

```
// get a list of all items in that folder
```

```
Var i,c as Integer
```

```
c=F.Count
```

```
for i=1 to c
```

```
files.Append f.TrueItem(i)
```

```
next
```

```
// delete each item
```

```
for each fo as FolderItem in files
```

```
if fo=nil then
```

```
' ignore
```

```
elseif fo.Directory then
```

```
deletefolder fo
```

```
fo.delete
```

```
else ' file
```

```
fo.Delete
```

```
end if
```

```
next
```

```
f.Delete
```

End Sub

See also:

- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1266
- 46.0.4 How to catch delete key? 1267
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.7 How to detect if CPU is 64bit processor? 1270
- 46.0.8 How to query variant type string for a variant? 1271
- 46.0.9 How to refresh a htmlviewer on Windows? 1272

46.0.7 How to detect if CPU is 64bit processor?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Via CPUID you can ask CPU:

Example:

```
Var c as new CPUIDMBS
```

```
if c.Flags(CPUIDMBS.kFeatureLM) then
MsgBox "64-bit CPU"
else
MsgBox "32-bit CPU"
end if
```

Notes: Should work on all intel compatible CPUs.

See also:

- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1266
- 46.0.4 How to catch delete key? 1267
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.6 How to delete a folder? 1269
- 46.0.8 How to query variant type string for a variant? 1271
- 46.0.9 How to refresh a htmlviewer on Windows? 1272

46.0.8 How to query variant type string for a variant?

Plugin Version: 20.5, Platforms: macOS, Linux, Windows.

Answer: The following example function returns type string for variant.

Example:

```
Public Function VariantTypeString(v as Variant) as string
// Xojo's VarType doesn't know Unsigned integers
'Dim type As Integer = VarType(v)

// MBS VarType can detect unsigned integer
Var type As Integer = GetVariantTypeMBS(v)

Var IsArray As Boolean = BitwiseAnd(type, Variant.TypeArray) = Variant.TypeArray

// type without array
type = BitwiseAnd(type, Bitwise.OnesComplement(Variant.TypeArray))

// build a dictionary to map types on first call
Static TypeMap As Dictionary
If TypeMap = Nil Then
TypeMap = New Dictionary
TypeMap.Value(Variant.TypeBoolean) = "Boolean"
TypeMap.Value(Variant.TypeCFStringRef) = "CFStringRef"
TypeMap.Value(Variant.TypeColor) = "Color"
TypeMap.Value(Variant.TypeCString) = "CString"
TypeMap.Value(Variant.TypeCurrency) = "Currency"
TypeMap.Value(Variant.TypeDate) = "Date"
TypeMap.Value(Variant.TypeDateTime) = "DateTime"
TypeMap.Value(Variant.TypeDouble) = "Double"
TypeMap.Value(Variant.TypeInt32) = "Int32"
TypeMap.Value(Variant.TypeInt64) = "Int64"
TypeMap.Value(Variant.TypeInteger) = "Integer"
TypeMap.Value(Variant.TypeNil) = "Nil"
TypeMap.Value(Variant.TypeObject) = "Object"
TypeMap.Value(Variant.TypeOSType) = "OSType"
TypeMap.Value(Variant.TypePString) = "PString"
TypeMap.Value(Variant.TypePtr) = "Ptr"
TypeMap.Value(Variant.TypeSingle) = "Single"
TypeMap.Value(Variant.TypeString) = "String"
TypeMap.Value(Variant.TypeStructure) = "Structure"
TypeMap.Value(Variant.TypeText) = "Text"
TypeMap.Value(Variant.TypeWindowPtr) = "WindowPtr"
TypeMap.Value(Variant.TypeWString) = "WString"

// MBS extra types
TypeMap.Value(Variant.TypeInt32+100) = "UInt32"
TypeMap.Value(Variant.TypeInt64+100) = "UInt64"
```

End If

```
// lookup type

#if DebugBuild then
If Not TypeMap.HasKey(type) Then
Break // missing type
End If
#endif

If IsArray Then
Return "Array of " + TypeMap.Lookup(type,"?")
Else
Return TypeMap.Lookup(type,"?")
End If
End Function
```

See also:

- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1266
- 46.0.4 How to catch delete key? 1267
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.6 How to delete a folder? 1269
- 46.0.7 How to detect if CPU is 64bit processor? 1270
- 46.0.9 How to refresh a htmlviewer on Windows? 1272

46.0.9 How to refresh a htmlviewer on Windows?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can ask the browser to reload the website with this code line:

Example:

```
call htmlViewer1.IERunJavaScriptMBS("javascript:document.location.reload()")
```

See also:

- 46.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1266
- 46.0.4 How to catch delete key? 1267
- 46.0.5 How to convert cmyk to rgb? 1268
- 46.0.6 How to delete a folder? 1269

- 46.0.7 How to detect if CPU is 64bit processor? 1273
 - 46.0.8 How to query variant type string for a variant? 1270
-
- 46.0.8 How to query variant type string for a variant? 1271

46.0.10 Is there an example for vector graphics in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this example inside the paint event of a window:

Example:

```
Var v as Group2D
Var r as RectShape
Var s as StringShape
```

```
const pi=3.14
```

```
s=new StringShape
s.Text="Hello World!"
s.TextFont="Geneva"
s.TextSize=24
s.FillColor=rgb(0,0,255)
s.Italic=true
s.y=5
s.x=0
```

```
r=new RectShape
```

```
r.X=0
r.y=0
r.Height=100
r.Width=180
r.BorderColor=rgb(255,0,0)
r.FillColor=rgb(0,255,0)
r.BorderWidth=5
r.Border=50
```

```
v=new Group2d
v.Append r
v.Append s
v.Rotation=pi*-20.0/180.0
v.x=150
v.y=150
```

```
g.DrawObject v
```

46.0.11 Picture functions do not preserve resolution values?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the picture functions return pictures with no/default resolution values.

Example:

```
Var l as Picture = LogoMBS(500)
```

```
l.HorizontalResolution = 300
```

```
l.VerticalResolution = 300
```

```
Var r as Picture = l.Rotate90MBS
```

```
MsgBox str(r.HorizontalResolution)+" x "+str(r.VerticalResolution)
```

```
r.HorizontalResolution = l.HorizontalResolution
```

```
r.VerticalResolution = l.VerticalResolution
```

```
MsgBox str(r.HorizontalResolution)+" x "+str(r.VerticalResolution)
```

Notes: So please fix them yourself after calling a function.

Maybe in the future this changes, but currently you can't really set this easily from plugin code.

46.0.12 A toolbox call needs a rect - how do I give it one?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Fill a memoryblock like this:

Example:

```
Var MB As Memoryblock
```

```
MB = NewMemoryBlock(8)
```

```
MB.Short(0) = window1.Top
```

```
MB.Short(2) = window1.Left
```

```
MB.Short(4) = window1.Height+window1.Top // bottom
```

```
MB.Short(6) = window1.Width+window1.Left // right
```

46.0.13 API client not supported?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you get this exception message on `SQLConnectionMBS.Connect`, we may have a problem.

Notes: First case is that the given thing is not supported (e.g. MS SQL directly on Mac).
 Second case is that the plugin compilation went wrong and the support for the database was not linked into the plugin. Like MySQL missing or MS SQL on Windows missing. In that case please contact us to fix the plugin.

46.0.14 Can I access Access Database with Java classes?

Plugin Version: all, Platform: Windows.

Answer: You can use ucanaccess to access databases created with Microsoft

Example:

```

Var options(-1) as string

// load all the jar files we have in a folder called java:

Var appFolder as FolderItem = GetFolderItem("")

Var count as Integer = appFolder.Parent.Child("java").Count
Var libjs() as string
For i as Integer = 1 to count
Var f As FolderItem = appFolder.Parent.Child("java").item(i)
If f <> Nil and f.Exists Then
libjs.append f.NativePath+";"
End If
Next

// now init virtual machine
Var library as string = Join(libjs, "")
Var vm as new JavaVMMBS(library)

if vm.Handle = 0 then
MsgBox "Failed to initialize virtual machine"
else
// now make a new database connection with ucanaccess
Var d as new JavaDatabaseMBS(vm,"net.ucanaccess.jdbc.UcanaccessDriver")
Var DbFile as FolderItem = appFolder.Parent.Child("Database11.accdb")
Var j as JavaConnectionMBS = d.getConnection("jdbc:ucanaccess://" + DbFile.NativePath)

// select and show values
Var r as JavaResultSetMBS = j.MySelectSQL("Select * From test")
while r.NextRecord
MsgBox r.getString("FirstName") + " " + r.getString("LastName")
wend

end if

```

Exception e as JavaExceptionMBS

```
MsgBox e.message+" errorcode: "+str(e.ErrorNumber)
```

Notes: see website:

<http://ucanaccess.sourceforge.net/site.html>

46.0.15 Can I create PDF from Xojo Report using DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, we have a graphics class integration for DynaPDF.

Notes: Since MBS Plugin in version 19.2, we can integrate reports with Xojo.

46.0.16 Can I use AppleScripts in a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, but they run on the server, not on the client.

Example:

Var a as new AppleScriptMBS

```
// query my application name
a.Compile "tell application ""System Events"" to return name of current application"
```

```
// run
a.Execute
```

```
// show result
label1.text = a.Result
```

```
// shows something like "My Application.fcgi.debug"
```

Notes: This can be useful to control the server from remote, if and only if the your sever is running Mac OS X.

46.0.17 Can I use graphics class with DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sorry, no. We can't provide a graphics subclass from plugin.

Notes: The is a feature request to allow graphics subclasses:

Feedback case 11391: [feedback://showreport?report_id=11391](https://feedback.apple.com/showreport?report_id=11391)

46.0.18 Can I use sockets on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, but they run on the server, not on the client.

Notes: You can use `HTTPSocket`, `SMTPSocket`, `POP3Socket`, `SMTPSecureSocket`, `SecurePOP3Socket`, `EasyTCPSocket`, `EasyUDPSocket`, `AutoDiscovery`, our Bonjour classes or our `CURL*` classes. But all of them work on the server, not on the client.

This means if you search for a printer with Bonjour, you can find the printers in the local network on your server hosting site. Using `SMTPSocket` may be a good idea for sending emails from the server like notifications.

46.0.19 Can I use your ChartDirector plugin on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, our ChartDirector plugin works just fine on the Xojo Web Edition.

Example:

```
// The data for the pie chart
Var data(-1) as Double=array(55.0, 18.0, 25.0, 22.0, 18.0, 30.0, 35.0)

// The labels for the pie chart, Words are choosen random to check font!
Var labels(-1) as string=array("Germany", "Italy", "France", "Spain", "UK", "Poland", "Russia")

// The colors to use for the sectors
Var colors(-1) as Integer

colors.Append &h66aaee
colors.Append &heebb22
colors.Append &hbbsbbb
colors.Append &h8844ff

if TargetLinux then
CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype/msttcorefonts"
end if

// Create a PieChart object of size 360 x 300 pixels
Var c as new CDPieChartMBS(700, 600)

c.setBackground(c.linearGradientColor(0, 0, 0, c.getHeight(), &h0000cc, &h000044))
```

```

c.setRoundedFrame(&hffffff, 16)
Var tt as CDTextBoxMBS = c.addTitle("ChartDirector Demonstration", "timesbi.ttf", 18)
tt.setMargin(0, 0, 16, 0)
tt.setFontColor(&hFFFFFF)

// Set the center of the pie at (180, 140) and the radius to 100 pixels
c.setPieSize 350,300,150
// Set the sector colors
c.setColors(c.kDataColor, colors)

// Draw the pie in 3D with a pie thickness of 20 pixels
c.set3D(20)

Var t as CDTextBoxMBS = c.setLabelStyle("arialbd.ttf", 10, &h000000)
t.setBackground(CDPieChartMBS.kSameAsMainColor, CDPieChartMBS.kTransparent, CDPieChartMBS.soft-
Lighting(CDPieChartMBS.kRight, 0))
t.setRoundedCorners(8)

// Use local gradient shading for the sectors, with 5 pixels wide
// semi-transparent white (bbffffff) borders
c.setSectorStyle(CDPieChartMBS.kLocalGradientShading, &hbbffffff, 0)

// Set the pie data and the pie labels
c.setData data,labels
call c.setLabelStyle "arialbd.ttf",18

Var pic as picture = c.makeChartPicture
Var wp as new WebPicture(pic, Picture.FormatJPEG) // JPEG makes it smaller and faster

ImageView1.Picture=wp

```

Notes: Be aware that our plugin produces pictures for you, which you assign to ImageViews. Transferring those pictures takes time, so you can optimize that with using WebPicture class. There you can decide between different compressions to improve speed (use JPEG instead of PNG).

e.g. if you use ubuntu, you can install the ttf-mscorefonts-installer package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

46.0.20 Can I use your DynaPDF plugin on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, our DynaPDF plugin works just fine on the Xojo Web Edition.

Notes: PDF files are created on the server. You may want to offer a preview to the user which uses reduced resolution images to reduce the time to download the PDF.

See our Create PDF example for the Xojo Web Edition.

46.0.21 Can I use your plugin controls on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: No.

46.0.22 Can you get an unique machine ID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: There is nothing like an unique machine ID.

Notes: 1:

You can use the MAC IDs of the network interfaces.

This can be changed by the user with software tools.

And the list of network interfaces changes if user reorder the interfaces.

2:

You can use the system folder creation date/time.

This may stay equal after cloning machines or after migration to new PC.

3:

You can use the Mac Serialnumber.

Mac only and it can happen that a Mac does not have a serial number.

4:

You can use the x86 CPU ID.

This is x86 CPU only and does not avoid running on the same CPU in different PCs.

46.0.23 ChartDirector: Alignment Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Alignment Specification

Notes: In many ChartDirector objects, you may specify the alignment of the object's content relative to its boundary. For example, for a TextBox object, you may specify the text's alignment relative to the box

boundary by using `TextBox.setAlignment`.

The `ChartDirector` API defines several constants for the alignment options.

ConstantValueDescription

<code>BottomLeft</code>	1	The leftmost point on the bottom line.
<code>BottomCenter</code>	2	The center point on the bottom line.
<code>BottomRight</code>	3	The rightmost point on the bottom line.
<code>Left</code>	4	The leftmost point on the middle horizontal line.
<code>Center</code>	5	The center point on the middle horizontal line.
<code>Right</code>	6	The rightmost point on the middle horizontal line.
<code>TopLeft</code>	7	The leftmost point on the top line.
<code>TopCenter</code>	8	The center point on the top line.
<code>TopRight</code>	9	The rightmost point on the top line.
<code>Bottom</code>	2	The center point on the bottom line. Same as <code>BottomCenter</code> .
<code>Top</code>	8	The center point on the top line. Same as <code>TopCenter</code> .
<code>TopLeft2</code>	10	An alternative top-left position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, <code>TopLeft2</code> refers to the left of the top side, while <code>TopLeft</code> refers to the top of the left side. The reverse applies for a horizontal axis.
<code>TopRight2</code>	11	An alternative top-right position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, <code>TopRight2</code> refers to the right of the top side, while <code>TopRight</code> refers to the top of the right side. The reverse applies for a horizontal axis.
<code>BottomLeft2</code>	12	An alternative bottom-left position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, <code>BottomLeft2</code> refers to the left of the bottom side, while <code>BottomLeft</code> refers to the bottom of the left side. The reverse applies for a horizontal axis.
<code>BottomRight2</code>	13	An alternative bottom-right position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, <code>BottomRight2</code> refers to the right of the bottom side, while <code>BottomRight</code> refers to the bottom of the right side. The reverse applies for a horizontal axis.

46.0.24 ChartDirector: Color Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: `ChartDirector: Color Specification`

Notes: Many functions in the `ChartDirector` API accept colors as parameters. `ChartDirector` supports colors specified in web and HTML compatible ARGB format, in which ARGB refers to the Alpha transparency, Red, Green and Blue components of the color.

In addition to ARGB colors, ChartDirector supports "dynamic" colors. A dynamic color is a color that changes depending on the position of the pixels. The "dynamic" colors that ChartDirector supports include "pattern colors", "metal colors", "gradient colors", "zone colors" and "dash line colors".

ChartDirector supports specifying colors indirectly using "palette colors". When a "palette color" is used, the color is specified as an index to a palette. The actual color is looked up from the palette. ARGB Color ARGB color consists of 4 components - alpha transparency, red, green and blue. The four components are encoded as a 32-bit number, with each component occupying 8 bits. In hexadecimal notation, it is AAR-RGGBB, where AA, RR, GG and BB are the alpha transparency, red, green and blue components.

Each component ranges from 00 - FF (0 - 255), representing its intensity. For example, pure red color is 00FF0000, pure green color is 0000FF00, and pure blue color is 000000FF. White color is 00FFFFFF, and black color is 00000000.

Most programming language requires you to put special prefix in front of hexadecimal characters. For C++, the prefix is "0x". For example, the syntax for the hexadecimal number 00FFFFFF is 0x00FFFFFF, or simply 0xFFFFFF.

For the alpha transparency component, a zero value means the color is not transparent at all. This is equivalent to traditional RGB colors. A non-zero alpha transparency means the color is partially transparent. The larger the alpha transparency, the more transparent the color will be. If a partially transparent color is used to draw something, the underlying background can still be seen.

For example, 80FF0000 is a partially transparent red color, while 00FF0000 is a non-transparent red color.

Note that ChartDirector's ARGB color is web and HTML compatible. For example, red is FF0000, the same as in HTML. There are many resources on the web that provide tables in which you can click a color and it will show its HTML color code. These color codes can be used in ChartDirector.

If alpha transparency is FF (255), the color is totally transparent. That means the color is invisible. It does not matter what the RGB components are. So in ChartDirector, only one totally transparent color is used - FF000000. All other colors of the form FFnnnnnn are reserved to represent palette colors and dynamic colors, and should not be interpreted as the normal ARGB colors.

The totally transparent color FF000000 is often used in ChartDirector to disable drawing something. For example, if you want to disable drawing the border of a rectangle, you can set the border color to totally transparent.

For convenience, ChartDirector defines a constant called Transparent, which is equivalent to FF000000. Pattern Color

A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

Pattern colors are created using `BaseChart.patternColor`, `BaseChart.patternColor2`, `DrawArea.patternColor` and `DrawArea.patternColor2`. The `patternColor` method creates pattern colors using an array of colors as a bitmap. The `patternColor2` method creates pattern colors by loading the patterns from image files.

These methods return a 32-bit integer acting as a handle to the pattern color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Metal Color
A metal color is a color of which the brightness varies smoothly across the chart surface as to make the surface look shiny and metallic. `ChartDirector` supports using any color as the base color of the metal color. In particular, using yellow and grey as the base colors will result in metal colors that look gold and silver.

Metal colors are most often used as background colors of charts. They are created using `CDBaseChartMBS.metalColor`, `CDBaseChartMBS.goldColor` and `CDBaseChartMBS.silverColor`. The first method allows you to specify an arbitrary base color. The second and third methods use yellow and grey as the base colors, resulting in gold and silver metal colors.

These methods return a 32-bit integer acting as a handle to the gradient color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Gradient Color
A gradient color is a color that changes progressively across a direction.

Gradient colors are created using `BaseChart.gradientColor`, `BaseChart.gradientColor2`, `DrawArea.gradientColor` and `DrawArea.gradientColor2`. The `gradientColor` method creates a 2-point gradient color that changes from color A to color B. The `gradientColor2` method creates a multi-point gradient colors that changes from color A to B to C

These methods return a 32-bit integer acting as a handle to the gradient color. The handle can be used in any `ChartDirector` API that expects a color as its input.

One common use of multi-point gradient colors is to define colors that have metallic look and feel. Please refer to `DrawArea.gradientColor2` for details.

Dash Line Colors
A dash line color is a color that switches on and off periodically. When used to draw a line, the line will appear as a dash line.

Dash line colors are created using `BaseChart.dashLineColor` and `DrawArea.dashLineColor`. They accept a line color and a dash pattern code as arguments, and return a 32-bit integer acting as a handle to the dash line color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Zone Colors
A zone color is for XY charts only. It is a color that automatically changes upon reaching a data threshold value along the x-axis or y-axis. Zone colors are created using `Layer.xZoneColor`, `Layer.yZoneColor`, `XYChart.xZoneColor` or `XYChart.yZoneColor`.

Palette Colors
Palette colors are colors of the format `FFFFnnnn`, where the least significant 16 bits (`nnnn`) are the index to the palette. A palette is simply an array of colors. For a palette color, the actual color is obtained by looking up the palette using the index. For example, the color `FFFF0001` is the second color in the palette (first color is index 0).

The colors in the palette can be ARGB colors or "dynamic" colors (pattern, gradient and dash line colors).

The first eight palette colors have special significance. The first three palette colors are the background color, default line color, and default text color of the chart. The 4th to 7th palette colors are reserved for future use. The 8th color is a special dynamic color that is equal to the data color of the "current data set".

The 9th color (index = 8) onwards are used for automatic data colors. For example, in a pie chart, if the sector colors are not specified, ChartDirector will automatically use the 9th color for the first sector, the 10th color for the second sector, and so on. Similarly, for a multi-line chart, if the line colors are not specified, ChartDirector will use the 9th color for the first line, the 10th color for the second line, and so on.

The ChartDirector API defines several constants to facilitate using palette colors.

ConstantValueDescription

Palette	FFFF0000	The starting point of the palette. The first palette color is (Palette + 0). The nth palette color is (Palette + n - 1).
BackgroundColor	FFFF0000	The background color.
LineColor	FFFF0001	The default line color.
TextColor	FFFF0002	The default text color.
[Reserved]	FFFF0003 - FFFF0006	These palette positions are reserved. Future versions of ChartDirector may use these palette positions for colors that have special significance.
SameAsMainColor	FFFF0007	A dynamic color that is equal to the data color of the current data set. This color is useful for objects that are associated with data sets. For example, in a pie chart, if the sector label background color is SameAsMainColor, its color will be the same as the corresponding sector color.
DataColor	FFFF0008	The starting point for the automatic data color allocation.

When a chart is created, it has a default palette. You may modify the palette using BaseChart.setColor, BaseChart.setColors, or BaseChart.setColors2.

The advantages of using palette colors are that you can change the color schemes of the chart in one place. ChartDirector comes with several built-in palettes represented by the following predefined constants.

ConstantDescription

defaultPalette	An array of colors representing the default palette. This palette is designed for drawing charts on white backgrounds (or lightly colored backgrounds).
whiteOnBlackPalette	An array of colors useful for drawing charts on black backgrounds (or darkly colored backgrounds).
transparentPalette	An array of colors useful drawing charts on white backgrounds (or lightly colored backgrounds). The data colors in this palette are all semi-transparent.

46.0.25 ChartDirector: Font Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Font Specification

Notes: Font Name

In ChartDirector, the font name is simply the file name that contains the font. For example, under the Windows platform, the "Arial" font is "arial.ttf", while the "Arial Bold" font is "arialbd.ttf".

NOTE: Mac OS X Specific Information

In Mac OS X, in addition to ".ttf", ChartDirector also supports Mac OS X font file formats, such as Font Suitcase files and Datafork files (.dfont). These files often contain multiple fonts. For example, the "GillSans.dfont" file contains 6 fonts.

So in addition to the file name, an index is needed to determine the font. The index is specified by appending a " | " character to the font name, followed by the index number. For example, the third font in "GillSans.dfont" is denoted as "GillSans.dfont | 2". (Note: The first font starts at 0.) If no index number is provided, the first font is assumed.

ChartDirector also supports using Mac OS X Font Manager names. For example, one may use "Gill Sans Light Italic" instead of using "GillSans.dfont | 1" as the font name. However, the Mac OS X Font Manager is active only if someone has logged into the Mac GUI console, so this method is only recommended for developing applications that run on the GUI console.

The sample programs that come with ChartDirector are designed to run on all operating systems, so they use generic font file names (eg. "arial.ttf") instead of Mac OS X specific names. To allow them to run on Mac OS X, ChartDirector on Mac OS X has a built-in table to map common font file names to Mac OS X font names:

"arial.ttf", "arialbd.ttf", "ariali.ttf" and "arialbi.ttf" are mapped to "Arial | 0" (Arial), "Arial | 1" (Arial Bold), "Arial | 2" (Arial Italic) and "Arial | 3" (Arial Bold Italic)

"times.ttf", "timesbd.ttf", "timesi.ttf" and "timesbi.ttf" are mapped to "Times New Roman | 0" (Times New Roman), "Times New Roman | 1" (Times New Roman Bold), "Times New Roman | 2" (Times New Roman Italic) and "Times New Roman | 3" (Times New Roman Bold Italic)

"cour.ttf", "courbd.ttf", "couri.ttf" and "courbi.ttf" are mapped to "Courier New | 0" (Courier New), "Courier New | 1" (Courier New Bold), "Courier New | 2" (Courier New Italic) and "Courier New | 3" (Courier New Bold Italic)

Font Location

ChartDirector on Windows does not come with any font files. It relies on the operating system's font files in the "[windows] \Fonts" directory. To see what fonts are installed in your operating system and their file names, use the File Explorer to view that directory.

ChartDirector on Windows will also search for the font files in the "fonts" subdirectory (if it exists) under the directory where the ChartDirector DLL "chartdir.dll" is installed. This is useful for private fonts. Also, for some especially secure web servers, the web anonymous user may not have access to the "[windows] \Fonts" directory. In this case, you may copy the font files to the above subdirectory.

ChartDirector on Mac OS X relies on operating system font files in "/Library/Fonts" and "/System/Library/Fonts".

ChartDirector on Linux, FreeBSD and Solaris assume the fonts files are in the "fonts" subdirectory under the directory where the ChartDirector shared object "libchartdir.so" is installed. ChartDirector on Linux, FreeBSD and Solaris come with a number of font files in the "fonts" subdirectory.

To keep the download size small, ChartDirector on Linux, FreeBSD and Solaris only come with some commonly used fonts. You may download additional fonts from the Internet. In particular, the Microsoft fonts at

http://sourceforge.net/project/showfiles.php?group_id=34153&release_id=105355

is highly recommended. Please refer to

<http://www.microsoft.com/typography/faq/faq8.htm>

on how you could use the fonts legally in your system.

ChartDirector supports True Type fonts (.ttf), Type 1 fonts (.pfa and .pfb) and Windows bitmap fonts (.fon). On Mac OS X, ChartDirector also supports Font Suitcase and Datafork (.dfont) files. On Linux, FreeBSD and Solaris, ChartDirector also supports Portable Compiled Fonts (.pcf fonts).

If you want ChartDirector to search other directories for the font files, you may list the directories in an environment variable called "FONTSPATH".

If you specify an absolute path name for the font file, ChartDirector will use the absolute path name and will not search other directories.

Artificial Boldening and Italicizing

Whereas most popular font comes with different styles for "normal", "bold", "italic" and "bold italic", some fonts only come with one style (the normal style). For example, the Monotype Corsiva font that comes with MS Office only has the normal style (mtcorsva.ttf). For these cases, you may append the "Bold" and/or "Italic" words after the font file name (separated with a space) to ask ChartDirector to artificially bolden and/or italicize the font. For example, you may specify the font name as "mtcorsva.ttf Bold".

Font List

Instead of specifying a single font file as the font name, you may specify a list of font files as the font name, separated by semi-colons. This is useful when using international characters that are only available in some fonts.

For example, if you would like to use the Arial font ("arial.ttf") for western characters, and the MingLiu font "mingliu.ttc" for Chinese characters (since the Arial font does not have Chinese characters), you may specify the font name as "arial.ttf;mingliu.ttc". In this case, ChartDirector will try the Arial font first. If it cannot find a certain character there, it will try the MingLiu font.

Indirect Font Names

ChartDirector supports several special keywords for specifying the font name indirectly. When these keywords are used as font names, ChartDirector will look up the actual font names from a font table. The keywords are as follows:

KeywordsDescription

"normal"	This default normal font, which is the first font in the font table. This is initially mapped to "arial.ttf" (Arial).
"bold"	The default bold font, which is the second font in the font table. This is initially mapped to "arialbd.ttf" (Arial Bold).
"italic"	The default italic font, which is the third font in the font table. This is initially mapped to "ariali.ttf" (Arial Italic).
"boldItalic"	The default bold-italic font, which is the fourth font in the font table. This is initially mapped to "arialbi.ttf" (Arial Bold Italic).
"fontN"	The (N + 1)th font in the font table (the first font is "font0").

The font table can be modified using `BaseChart.setFontTable` or `DrawArea.setFontTable`.

The advantage of using indirect font names is that you can change the fonts in your charts in one place.

Font Index
Most font files contain one font. However, it is possible a font file contains multiple fonts (that is, a font collection). For example, in True Type fonts, font files with extension ".ttc" may represent a font collection.

If a font file contains multiple font, the font index can be used to specify which font to use. By default, the font index is 0, which means the first font in the font file will be used.

Font Size
The font size decides how big a font will appear in the image. The font size is expressed in a font unit called points. This is the same unit used in common word processors.

Instead of specifying font size, some ChartDirector API (eg. `TextBox.setFontSize`) allow you to specify font height and font width separately. You may use different point sizes for font height and font width to create special effects.

Font Color
This is the color to draw the font. (See [Color Specification](#) on how colors are represented in ChartDirector.)

Font Angle
This is the angle in degrees by which the font should be rotated anti-clockwise.

Vertical Layout
By default, text are laid out horizontally, with characters being drawn from left to right.

ChartDirector also supports vertical layout, with characters being drawn from top to bottom. For example, you may use `BaseChart.addText` to add text that are laid out vertically. Vertical layout is common for oriental languages such as Chinese, Japanese and Korean.

46.0.26 ChartDirector: Mark Up Language

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Mark Up Language

Notes: ChartDirector Mark Up Language (CDML) is a language for including formatting information in text strings by marking up the text with tags.

CDML allows a single text string to be rendered using multiple fonts, with different colors, and even embed images in the text. **Font Styles**

You can change the style of the text by using CDML tags. For example, the line:

```
<*font=timesi.ttf,size=16,color=FF0000>Hello <*font=arial.ttf,size=12,color=8000*>world!
```

will result in the following text rendered:

In general, all tags in CDML are enclosed by `<*` and `*>`. Attributes within the tags determine the styles of the text following the tags within the same block.

If you want to include `<*` in text without being interpreted as CDML tags, use `«*` as the escape sequence.

The following table describes the supported font style attributes in CDML. See [Font Specification](#) for details on various font attributes.

AttributeDescription

font	Starts a new style section, and sets the font name. You may use this attribute without a value (that is, use "font" instead of "font=arial.ttf") to create a new style section without modifying the font name.
size	The font size.
width	The font width. This attribute is used to set the font width and height to different values. If the width and height are the same, use the size attribute.
height	The font height. This attribute is used to set the font width and height to different values. If the width and height are the same, use the size attribute.
color	The text color in hex format.
bgColor	The background color of the text in hex format.
underline	The line width of the line used to underline the following characters. Set to 0 to disable underline.
sub	Set the following text to be in subscript style. This attribute does not need to have a value. (You may use "sub" as the attribute instead of "sub=1".)
super	Set the following text to be in superscript style.

Set the following text to be in superscript style. This attribute does not need to have a value. (You may use "super" as the attribute instead of "super=1".)

xoffset	Draw the following the text by shifting the text horizontally from the original position by the specified offset in pixels.
yoffset	Draw the following the text by shifting the text vertically from the original position by the specified offset in pixels.
advance	Move the cursor forward (to the right) by the number of pixels as specified by the value this attribute.
advanceTo	Move the cursor forward (to the right) to the position as specified by the value this attribute. The position is specified as the number of pixels to the right of the left border of the block. If the cursor has already passed through the specified position, the cursor is not moved.

Note that unlike HTML tags, no double or single quotes are used in the tags. It is because CDML tags are often embedded as string literals in source code. The double or single quotes, if used, will conflict with the string literal quotes in the source code. Therefore in CDML, no quotes are necessary and they must not be used.

Also, unlike HTML tags, CDML uses the comma character as the delimiter between attributes. It is because certain attributes may contain embed spaces (such as the font file name). So space is not used as the delimiter and the comma character is used instead.

Note the font attribute above starts a new style section, while other attributes just modify the current style section. You may use `<*/font*>` to terminate a style section, which will restore the font styles to the state before the style section.

Blocks and Lines
In CDML, a text string may contain multiple blocks. A block may contain multiple lines of text by separating them with new line characters ("`\n`") or with `<*br*>`. The latter is useful for programming languages that cannot represent new line characters easily.

For example, the line:

```
<*size=15*><*block*><*color=FF*>BLOCK<*br*>ONE<*/*>and <*block*><*color=FF00*>BLOCK<*br*>TWO
```

will result in the following text rendered:

The above example contains a line of text. The line contains two blocks with the characters " and " in between. Each block in turn contains two lines. The blocks are defined using `<*block*>` as the start tag and `<*/*>` as the end tag.

When a block ends, font styles will be restored to the state before entering the block.
Embedding Images
CDML supports embedding images in text using the following syntax:

`<*img=my_image_file.png*>`

where `my_image_file.png` is the path name of the image file.

For example, the line:

`<*size=20*>A <*img=sun.png*>day`

will result in the following text rendered:

ChartDirector will automatically detect the image file format using the file extension, which must either `png`, `jpg`, `jpeg`, `gif`, `wbmp` or `wmp` (case insensitive).

Please refer to `BaseChart.setSearchPath` or `DrawArea.setSearchPath` on the directory that ChartDirector will search for the file.

The `<*img*>` tag may optionally contain width and height attributes to specify its pixel width and height. In this case, ChartDirector will stretch or compress the image if necessary to the required width and height.

CDML supports nesting blocks, that is, a block can contain other sub-blocks. Attributes are supported in the `<*block*>` tag to control the alignment and orientation of the sub-blocks. The `<*img=my_image_file.png*>` is treated as a block for layout purposes.

For example, the line:

`<*block,valign=absmiddle*><*img=molecule.png*><*block*>Hydrazino\nMolecule<*/*><*/*>`

will result in the following text rendered:

The the above starts `<*block,valign=absmiddle*>` which specifies its content should align with each others in the vertical direction using the absolute middle alignment. The block contains an image, followed by a space characters, and then another block which has two lines of text.

The following table describes the supported attributes inside `<*block*>` tag:

AttributeDescription

The value `baseline` means the baseline of sub-blocks should align with the baseline of the block. The baseline is the underline position of text. This is normal method of aligning text, and is the default in CDML. For images or blocks that are rotated, the baseline is the same as the bottom.

The value `top` means the top line of sub-blocks should align with the top line of the block.

width	The width of the block in pixels. By default, the width is automatically determined to be the width necessary for the contents of the block. If the width attribute is specified, it will be used as the width of the block. If the width is insufficient for the contents, the contents will be wrapped into multiple lines.
height	The height of the block in pixels. By default, the height is automatically determined to be the height necessary for the contents of the block. If the height attribute is specified, it will be used as the height of the block.
maxwidth	The maximum width of the block in pixels. If the content is wider than maximum width, it will be wrapped into multiple lines.
truncate	The maximum number of lines of the block. If the content requires more than the maximum number of lines, it will be truncated. In particular, if truncate is 1, the content will be truncated if it exceeds the maximum width (as specified by maxwidth or width) without wrapping. The last few characters at the truncation point will be replaced with "...".
linespacing	The spacing between lines as a ratio to the default line spacing. For example, a line spacing of 2 means the line spacing is two times the default line spacing. The default line spacing is the line spacing as specified in the font used.
bgColor	The background color of the block in hex format.
valign	The vertical alignment of sub-blocks. This is for blocks that contain sub-blocks. Supported values are baseline, top, bottom, middle and absmiddle.

The value bottom means the bottom line of sub-blocks should align with the bottom line of the block.

The value middle means the middle line of sub-blocks should align with the the middle line of the block. The middle line is the middle position between the top line and the baseline.

The value absmiddle means the absolute middle line of sub-blocks should align with the absolute middle line of the block. The absolute middle line is the middle position between the top line and the bottom line.

halign The horizontal alignment of lines. This is for blocks that contain multiple lines. Supported values are left, center and right.

The value left means the left border of each line should align with the left border of the block. This is the default.

The value center means the horizontal center of each line should align with the horizontal center of the block.

The value right means the right border of each line should align with the right border of the block.

angle Rotate the content of the block by an angle. The angle is specified in degrees in counter-clockwise direction.

46.0.27 ChartDirector: Parameter Substitution and Formatting

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Parameter Substitution and Formatting

Notes: ChartDirector charts often contain a lot of text strings. For example, sector labels in pie charts, axis labels for x and y axes, data labels for the data points, HTML image maps, etc, are all text strings.

ChartDirector uses parameter substitution to allow you to configure precisely the information contained in the text and their format.

Format Strings

In parameter substitution, format strings are used to specify the entities to be include into labels and how to format numbers and dates.

For example, when drawing a pie chart with side label layout, the default sector label format string is:

```
" { label } ( { percent } %)"
```

When the sector label is actually drawn, ChartDirector will replace " { label } " with the sector name, and " { percent } " with the sector percentage. So the above label format will result is a sector label similar to "ABC (34.56%)".

You may change the sector label format by changing the format string. For example, you may change it to:

```
" { label } : US$ { value | 2 } K ( { percent } %)"
```

The sector label will then become something like "ABC: US\$ 123.00 (34.56%)".

In general, in ChartDirector parameter substitution, parameters enclosed by curly brackets will be substituted with their actual values when creating the texts.

For parameters that are numbers or dates/times, ChartDirector supports a special syntax in parameter substitution to allow formatting for these values. Please refer to the Number Formatting and Date/Time Formatting sections below for details.

Parameter Expressions

ChartDirector supports numeric expressions in format strings. They are denoted by enclosing the expression with curly brackets and using "=" as the first character. For example:

```
"USD { value } (Euro { = { value } *0.9 } )"
```

In the above, " { value } " will be substituted with the actual value of the sector. The expression " { = { value } *0.9 } " will be substituted with the actual value of the sector multiplied by 0.9.

ChartDirector parameter expressions support operators "+", "-", "*", "/", "%" (modulo) and "^" (exponentiation). Operators "*", "/", "%", "^" is computed first, followed by "+" and "-". Operators of the same precedence are computed from left to right). Parenthesis "(" and ")" can be used to change the computation order.

Parameters for Pie Charts

The following table describes the parameters available for pie charts.

Parameter	Description
sector	The sector number. The first sector is 0, while the nth sector is (n-1).
dataSet	Same as { sector } . See above.
label	The text label of the sector.
dataSetName	Same as { label } . See above.
value	The data value of the sector.
percent	The percentage value of the sector.
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using BaseChart.addExtraField or BaseChart.addExtraField2.

Parameters for All XY Chart Layers

The followings are parameters that are apply to all XY Chart layers in general. Some layer types may have additional parameters (see below).

Note that certain parameters are inapplicable in some context. For example, when specifying the aggregate label of a stacked bar chart, the { dataSetName } parameter is inapplicable. It is because a stacked bar is composed of multiple data sets. It does not belong to any particular data set and hence does not have a data set name.

{ fieldN } means the extra field is indexed by the data point number. The Pth data point corresponds to the Pth element of the extra field.

Additional Parameters for Line Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Trend Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Box-Whisker Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for HLOC and CandleStick Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Vector Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Parameters for All Polar Layers

The followings are parameters that are apply to all Polar Chart layers in general. Some layer types may have additional parameters (see below).

{ fieldN } means the extra field is indexed by the data point number. The Pth data point corresponds to the Pth element of the extra field.

Additional Parameters for PolarVector Layers

The followings are parameters that are in additional to the parameters for all Polar Chart layers.

Parameters for Axis

The following table describes the parameters available for pie charts.

Number Formatting

For parameters that are numbers, ChartDirector supports a number of formatting options in parameter substitution.

For example, if you want a numeric field { value } to have a precision of two digits to the right of the decimal point, use ',' (comma) as the thousand separator, and use '.' (dot) as the decimal point, and you may use { value | 2, . } . The number 123456.789 will then be displayed as 123,456.79.

For numbers, the formatting options are specified using the following syntax:

```
{ [ param ] | [ a ] [ b ] [ c ] [ d ] }
```

where:

If this field starts with "E" or "e", followed by a number, it means formatting the value using scientific notation with the specified number of decimal places. If the "E" or "e" is not followed by a number, 3 is assumed.

For example, { value | E4 } will format the value 10.3 to 1.0300E+1, and { value | e4 } will format the same value to 1.0300e+1.

If this field starts with "G" or "g", followed by a number, it means formatting the value using the scientific notation only if the value is large and requires more than the specified number of digits, or the value is less than 0.001. If scientific notation is used, the number following "G" or "g" also specifies the number of significant digits to use. If the "G" or "g" is not followed by a number, 4 is assumed.

For example, consider the format string { value | G4 } . The value 10 will be formatted to 10. The value 100000 will be formatted to 1.000E+5. Similarly, for { value | g4 } , the value 10 will be formatted to 10, while the value 100000 will be formatted to 1.000e+5.

If you skip this argument, ChartDirector will display the exact value using at most 6 decimal places.

You may skip [b] [c] [d] . In this case, the default will be used.

Date/Time Formatting

For parameters that are dates/times, the formatting options can be specified using the following syntax:

```
{ [ param ] | [ datetime_format_string ] }
```

where [datetime_format_string] must start with an english character (A-Z or a-z) that is not "G", "g", "E" or "e", and may contain any characters except ' ' . (If it starts with "G", "g", "E" or "e", it will be considered as a number format string.)

Certain characters are substituted according to the following table. Characters that are not substituted will be copied to the output.

For example, a parameter substitution format of { value | mm-dd-yyyy } will display a date as something similar to 09-15-2002. A format of { value | dd/mm/yy hh:nn:ss a } will display a date as something similar to 15/09/02 03:04:05 pm.

If you want to include characters in the format string without substitution, you may enclose the characters in single or double quotes.

For example, the format { value | mmm '<*color=dd0000*>'yyyy } will display a date as something like Jan <*color=dd0000*>2005 (the <*color=dd0000*> is a CDML tag to specify red text color). Note that the <*color=dd0000*> tag is copied directly without substitution, even it contains "dd" which normally will

be substituted with the day of month.

Escaping URL/HTML/CDML characters

Parameter substitution is often used to create HTML image maps. In HTML, some characters has special meanings and cannot be used reliably. For example, the '>' is used to represent the end of an HTML tag.

Furthermore, if the field happens to be used as an URL, characters such as '?', '&' and '+' also have special meanings.

By default, ChartDirector will escape template fields used in URL and query parameters when generating image maps. It will modify URL special characters to the URL escape format "%XX" (eg. "?" will become "%3F"). After that, it will modify HTML special characters to the HTML escape format "&#nn;" (eg. ">" will become ">"). Similarly, it will escape other attributes in the image map using HTML escape format (but not URL escape format).

In addition to escaping HTML and URL special characters, ChartDirector will also remove CDML fields in creating image maps. It is because CDML is only interpreted in ChartDirector, should not be useful outside of ChartDirector (such as in browser tool tips).

In some cases, you may not want ChartDirector to escape the special characters. For example, if the parameters have already been escaped before passing to ChartDirector, you may want to disable ChartDirector from escaping them again.

ChartDirector supports the following special fields to control the escape methods - "{ escape_url }", "{ noescape_url }", "{ escape_html }", "{ noescape_html }", "{ escape_cdml }" and "{ noescape_cdml }". These fields enable/disable the escape methods used in the template fields that follow them.

46.0.28 ChartDirector: Shape Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Shape Specification

Notes: Several ChartDirector API accept shape specification as arguments. For example, BarLayer.setBarShape and BarLayer.setBarShape2 can be used to specify shapes of bars in bar charts, while DataSet.setDataSymbol, DataSet.setDataSymbol4, PolarLayer.setDataSymbol and PolarLayer.setDataSymbol4 can be used to specify shapes for data symbols.

Note that in addition to shapes, in many cases ChartDirector also accepts images or custom draw objects for data representation. For example, see DataSet.setDataSymbol2, DataSet.setDataSymbol3, PolarLayer.setDataSymbol2 and PolarLayer.setDataSymbol3.

Built-In Shapes

Built-in shapes are specified as integers. The integers can be explicit constants, or can be generated by a `ChartDirector` method for parameterized shapes. For example, a circle is represented by an explicit constant `CircleShape (=7)`. On the other hand, the number representing a polygon depends on the number of sides the polygon has, so it is generated by using the `PolygonShape` method, passing in the number of sides as argument.

The following table illustrates the various `ChartDirector` shapes:

Custom Shapes

In `ChartDirector`, custom shapes are specified as an array of integers `x0, y0, x1, y1, x2, y2 ...` representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 1000 x 1000 units, in which the x-axis is from -500 to 500 going from left to right, and the y-axis is from 0 to 1000 going from bottom to top.

`ChartDirector` will automatically scale the polygon so that 1000 units will become to the pixel size as requested by the various `ChartDirector` API.

As an example, the shape of the standard diamond shape in `ChartDirector` is represented as an array with 8 numbers:

```
0, 0, 500, 500, 0, 1000, -500, 500
```

46.0.29 Copy styled text?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: How to quickly copy styled text from one textarea to another?

Example:

```
#if TargetWin32 then
TextArea1.WinRTFDataMBS = TextArea2.WinRTFDataMBS
#elseif TargetMacOS then
TextArea1.NSTextViewMBS.textStorage.setAttributedString TextArea2.NSTextViewMBS.textStorage
#else
TextArea1.StyledText = TextArea2.StyledText
#endif
```

Notes: The code above uses special plugin functions on Mac and Windows and falls back to framework for Linux.

46.0.30 Do you have code to validate a credit card number?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can check the checksum to tell if a credit card number is not valid.

Example:

```

Var strNumber As String
Var nLength as Integer
Var nValue as Integer
Var nChecksum as Integer
Var nIndex as Integer

strNumber = EditField1.Text
nLength = Len(strNumber)
nChecksum = 0

For nIndex = 0 To nLength - 2
nValue = Val(Mid(strNumber, nLength - (nIndex + 1), 1)) * (2 - (nIndex Mod 2))
If nValue <10 Then
nChecksum = nChecksum + nValue
Else
nChecksum = nChecksum + (nValue - 9)
End If
Next

If Val(Mid(strNumber, Len(strNumber), 1)) = (10 - (nChecksum Mod 10)) Mod 10 Then
MsgBox("The credit card number looks valid")
Else
MsgBox("The credit card number is invalid")
End IF

```

Notes: Here's some code that will validate the checksum for a credit card. It works for Visa, MasterCard, American Express and Discover. Not sure about others, but I imagine they use the same basic algorithm. Of course, this doesn't actually mean that the credit card is valid, it's only useful for helping the user catch typos.

The above code doesn't have any error checking and it expects that the credit card number will be entered without spaces, dashes or any other non-numeric characters. Addressing those issues will be an exercise left to the reader. :)

(From Mike Stefanik)

46.0.31 Do you have plugins for X-Rite EyeOne, eXact or i1Pro?

Plugin Version: all.

Answer: Our EyeOne plugin is available on request for licensees of the X-Rite SDKs.

Notes: Please first go to X-Rite and get a SDK license.
Then we can talk about the plugin.

46.0.32 Does SQL Plugin handle stored procedures with multiple result sets?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the plugin can work with multiple recordsets.

Notes: You need to use SQLCommandMBS class. When you get back results, you use FetchNext to walk over all records in the first result set. Then you simply start again with FetchNext to get the second record set.

Even the RecordSet functions should work, just use them twice to get all records from both record sets.

46.0.33 Does the plugin home home?

Plugin Version: all, Platform: macOS.

Answer: Yes, we like to know who is using the plugin, so the plugin may contact our server.

Example:

none.

Notes: Please note that this does not affect your users as the plugin will only do this in the IDE and the relevant plugin part is never included in your applications.

The plugin if used for some hours, does contact our server to provide statistical data about Xojo version and OS versions. This way we know what versions are used. We can return the version number of the current plugin which may be visible in future versions somehow. And we transmit partial licenses data so we can track use of illegal license keys.

If you do not like to have this, you can block Xojo IDE from contacting our website via your Firewall.

Blocking the transfer will not disable the plugin or change the features.

Or contact us for a plugin version which explicitly does not contain this feature.

46.0.34 folderitem.absolutePath is limited to 255 chars. How can I get longer ones?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Paths on a Mac are not unique, so use them only to display them to the user.

Example:

```

Function AbsolutePath(f as FolderItem) As String
Var s as string
Var nf as FolderItem
nf = f
s = ""
while nf<>nil
s = nf.name + "." + s
nf = nf.parent
wend
Return s
End Function

```

46.0.35 Has anyone played round with using CoreImage to do things like add dissolve transitions say when changing from one tab to another within a window?

Platform: macOS.

Answer: This code implements animations for a tabpanel change:

Example:

```

// in a tabpanel.change event:

Var r as CGSTransitionRequestMBS
Var co as new CGSConnectionMBS
Var cw as CGSWindowMBS
Var ct as CGSTransitionMBS
static OldTab as Integer

cw=co.CGSWindow(window1)
If cw = Nil Then
return // 10.3...
End If
r=new CGSTransitionRequestMBS
r.TransitionType=r.CGSFlip
r.HasBackGround=false
r.HasBackColor=false
r.Win=cw
// watch the value of the clicked tab versus the last tab
if tabpanel1.Value=0 or tabpanel1.Value <OldTab then
r.TransitionOption=r.CGSLeft
ct=co.NewTransition(r)
if ct<>Nil then
Refresh
ct.Invoke(1)

```

```

ct.Wait(1)
ct.Release
else
MsgBox "Error creating the transition."
end if
else
r.TransitionOption=r.CGSRight
ct=co.NewTransition(r)
if ct<>Nil then
Refresh
ct.Invoke(1)
ct.Wait(1)
ct.Release
else
MsgBox "Error creating the transition."
end if
end if
// Keep track of the last tab clicked
OldTab = tabpanel1.Value

```

Notes: See CGS* classes for more details.

46.0.36 How about Plugin support for older OS X?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We support in general Mac OS X 10.5 and newer.

Notes: All the 64-bit plugins on Mac require OS X 10.7.

Intel 32-bit plugins on Mac require OS X 10.5 or newer.

Currently the ChartDirector 6, GraphicsMagick and GameKit plugins requires Mac OS X 10.6.
Also for SQL Plugin the built in SQLite library requires 10.6.

46.0.37 How can I detect whether an Intel CPU is a 64bit CPU?

Plugin Version: all.

Answer: Look on the CPU family returned by sysctl:

Example:

Function is64bit() As Boolean

```
#if TargetLittleEndian
```

```

Var m as MemoryBlock = NewMemoryBlock(8)
Var family as Integer
Var s as string

m=SystemControlNameToMIBMBS("hw.cpubfamily")
m=SystemControlMBS(m)

if m<>nil then
m.LittleEndian=True
family=m.Long(0)

const CPUFAMILY_INTEL_6_14 = &h73d67300 /* "Intel Core Solo" and "Intel Core Duo" (32-bit
Pentium-M with SSE3) */
const CPUFAMILY_INTEL_6_15 = &h426f69ef /* "Intel Core 2 Duo" */
const CPUFAMILY_INTEL_6_23 = &h78ea4fbc /* Penryn */
const CPUFAMILY_INTEL_6_26 = &h6b5a4cd2 /* Nehalem */

Select case family
case CPUFAMILY_INTEL_6_14
Return false
case CPUFAMILY_INTEL_6_15
Return true
case CPUFAMILY_INTEL_6_23
Return true
case CPUFAMILY_INTEL_6_26
Return true
// newer CPUs may be missing here
end Select

end if

#endif

Return false

Exception
Return false
End Function

```

Notes: This code is written for Mac OS X where you only have a limited number of possible CPUs.

46.0.38 How can I disable the close box of a window on Windows?

Plugin Version: all, Platform: Windows.

Answer: The following code will remove the close item from the system menu of the window.

Example:

```

#if TargetWin32 then
Declare Function GetSystemMenu Lib "user32" (hwnd as Integer, bRevert as Integer) as Integer
Declare Function RemoveMenu Lib "user32" (hMenu as Integer, nPosition as Integer, wFlags as Integer) as Integer
Var hSysMenu as Integer
hSysMenu = GetSystemMenu(me.WinHWND, 0)
RemoveMenu hSysMenu, &HF060, &H0
#endif

```

Notes: The window may not be updated directly.

46.0.39 How can I get all the environment variables from Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```

#if targetWin32
declare function GetEnvironmentStrings Lib "kernel32" () as ptr
Var m as memoryBlock
Var n as Integer

m=GetEnvironmentStrings()

n=0
do
msgBox m.cstring(n)
while m.byte(n)<>0
n=n+1
wend
n=n+1
loop until m.byte(n)=0
#endif

```

Notes: The MBS Plugin has an EnvironmentMBS class for this.

46.0.40 How can i get similar behavior to Roxio Toast or iTunes where clicking a 'burn' button allows the next inserted blank CD-R to bypass the Finder and be accepted by my application?

Plugin Version: all, Platform: macOS.

Answer: You need to get a media reservation.

Example:

```
Var d as DRDeviceMBS // get a device
d.AcquireMediaReservation
```

Notes: Use the plugin function AcquireMediaReservation and later release it using ReleaseMediaReservation.

See plugin examples on how to use it and check Apples DiscRecording framework documentation for more details.

46.0.41 How can I get text from a PDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Crossplatform you can use DynaPDF Pro.

Notes: On Mac OS X you can also use PDFKit for the same job.

While DynaPDF Pro gives you each bit of text with rotation, font information and encoding details, PDFKit gives you only the text string for a PDF page.

46.0.42 How can I get text from a Word Document?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: to get the text string from a doc file, use the NSAttributedStringMBS class.

Notes: The NSAttributedStringMBS class is Mac OS X only and we have currently no solution for Windows or Linux.

Use the NSAttributedStringMBS.initWithDocFormat(data as string) as boolean method.

46.0.43 How can I get the item string for a given file creator?

Plugin Version: all.

Answer: Try this function:

Example:

```

Sub pullNativeDocs(aCREA As string)
  Var result as Integer
  Var m, k as memoryBlock
  Var f as folderItem
  Var newType as string
  Var anIcon As picture
  Var ofs as Integer

  Declare Function GetFileTypesThatAppCanNativelyOpen Lib "Carbon" (appVRefNumHint as Short, appSignature as OSType, nativeTypes as Ptr) as Short Inline68K("701CABFC")
  Declare Function GetDocumentKindString Lib "Carbon" (docVRefNum as Short, docType as OSType, docCreator as OSType, kindString as ptr) as Short Inline68K("7016ABFC")

  listBox1.deleteAllRows

  m = newMemoryBlock(1024)
  result = GetFileTypesThatAppCanNativelyOpen(Volume(0).MacVRefNum, aCREA, m)
  if result <> 0 then
    listBox1.addRow "<Not found.>"
  return
end if

do
  if m.byte(ofs*4) = 0 then
    exit
  else
    newType = m.OSTypeMBS(ofs*4)
    listBox1.addRow newType
    k = newMemoryBlock(64)
    result = GetDocumentKindString(Volume(0).MacVRefNum, newType, aCREA, k)
    if result = 0 then
      listBox1.cell(ofs,1) = k.pString(0)
      ofs = ofs + 1
    else
      listBox1.cell(ofs,1) = "(unknown)"
    end if
  end if
end if
loop

End Sub

```

Notes: Change "Translation" to "CarbonLib" for Mac OS X.

46.0.44 How can I launch an app using its creator code?

Plugin Version: all, Platform: macOS.

Answer: Send an AppleEvent "odoc" with the creator code to the Finder ("MACS"):

Example:

```
Function LaunchByCreator(C As String) As Boolean
Var A As AppleEvent
A = NewAppleEvent("aevt","odoc","MACS")
A.ObjectSpecifierParam("—") = GetUniqueIDObjectDescriptor("appf",nil,C)
return A.Send
End Function
```

46.0.45 How can I learn what shared libraries are required by a plugin on Linux?

Plugin Version: all, Platform: macOS.

Answer: Please use the ldd command in the terminal.

Notes: You build an app on any platform, but for Linux.

For the resulting .so files in the libs folder, you can run the ldd command with the library path as parameter. It shows you references lib files and you can make sure you have those installed.

This is a sample run of our graphicsmagick plugin:

```
cs@Ubuntu32:
textasciitilde /MeinProgramm/MeinProgramm Libs$ ldd libMBSGraphicsMagickPlugin17744.so
linux-gate.so.1 =>(0xb76ee000)
libdl.so.2 =>/lib/i386-linux-gnu/libdl.so.2 (0xb6f0e000)
libgtk-x11-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgtk-x11-2.0.so.0 (0xb6aa6000)
libpthread.so.0 =>/lib/i386-linux-gnu/libpthread.so.0 (0xb6a8a000)
libstdc++.so.6 =>/usr/lib/i386-linux-gnu/libstdc++.so.6 (0xb69a5000)
libm.so.6 =>/lib/i386-linux-gnu/libm.so.6 (0xb6979000)
libgcc_s.so.1 =>/lib/i386-linux-gnu/libgcc_s.so.1 (0xb695b000)
libc.so.6 =>/lib/i386-linux-gnu/libc.so.6 (0xb67b1000)
/lib/ld-linux.so.2 (0xb76ef000)
libgdk-x11-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgdk-x11-2.0.so.0 (0xb6701000)
libpangocairo-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpangocairo-1.0.so.0 (0xb66f4000)
libX11.so.6 =>/usr/lib/i386-linux-gnu/libX11.so.6 (0xb65c0000)
libXfixes.so.3 =>/usr/lib/i386-linux-gnu/libXfixes.so.3 (0xb65ba000)
libatk-1.0.so.0 =>/usr/lib/i386-linux-gnu/libatk-1.0.so.0 (0xb659a000)
libcairo.so.2 =>/usr/lib/i386-linux-gnu/libcairo.so.2 (0xb64ce000)
libgdk_pixbuf-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgdk_pixbuf-2.0.so.0 (0xb64ad000)
libgio-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgio-2.0.so.0 (0xb6356000)
libpangoft2-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpangoft2-1.0.so.0 (0xb632a000)
libpango-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpango-1.0.so.0 (0xb62e0000)
```

```

libfontconfig.so.1 =>/usr/lib/i386-linux-gnu/libfontconfig.so.1 (0xb62ab000)
libgobject-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgobject-2.0.so.0 (0xb625c000)
libglib-2.0.so.0 =>/lib/i386-linux-gnu/libglib-2.0.so.0 (0xb6163000)
libXext.so.6 =>/usr/lib/i386-linux-gnu/libXext.so.6 (0xb6151000)
libXrender.so.1 =>/usr/lib/i386-linux-gnu/libXrender.so.1 (0xb6147000)
libXinerama.so.1 =>/usr/lib/i386-linux-gnu/libXinerama.so.1 (0xb6142000)
libXi.so.6 =>/usr/lib/i386-linux-gnu/libXi.so.6 (0xb6132000)
libXrandr.so.2 =>/usr/lib/i386-linux-gnu/libXrandr.so.2 (0xb6129000)
libXcursor.so.1 =>/usr/lib/i386-linux-gnu/libXcursor.so.1 (0xb611e000)
libXcomposite.so.1 =>/usr/lib/i386-linux-gnu/libXcomposite.so.1 (0xb611a000)
libXdamage.so.1 =>/usr/lib/i386-linux-gnu/libXdamage.so.1 (0xb6115000)
libfreetype.so.6 =>/usr/lib/i386-linux-gnu/libfreetype.so.6 (0xb607b000)
libxcb.so.1 =>/usr/lib/i386-linux-gnu/libxcb.so.1 (0xb605a000)
libpixmap-1.so.0 =>/usr/lib/i386-linux-gnu/libpixmap-1.so.0 (0xb5fc2000)
libpng12.so.0 =>/lib/i386-linux-gnu/libpng12.so.0 (0xb5f98000)
libxcb-shm.so.0 =>/usr/lib/i386-linux-gnu/libxcb-shm.so.0 (0xb5f93000)
libxcb-render.so.0 =>/usr/lib/i386-linux-gnu/libxcb-render.so.0 (0xb5f89000)
libz.so.1 =>/lib/i386-linux-gnu/libz.so.1 (0xb5f73000)
libgmodule-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgmodule-2.0.so.0 (0xb5f6e000)
libselinux.so.1 =>/lib/i386-linux-gnu/libselinux.so.1 (0xb5f4f000)
libresolv.so.2 =>/lib/i386-linux-gnu/libresolv.so.2 (0xb5f36000)
libexpat.so.1 =>/lib/i386-linux-gnu/libexpat.so.1 (0xb5f0c000)
libffi.so.6 =>/usr/lib/i386-linux-gnu/libffi.so.6 (0xb5f05000)
libpcre.so.3 =>/lib/i386-linux-gnu/libpcre.so.3 (0xb5ec9000)
librt.so.1 =>/lib/i386-linux-gnu/librt.so.1 (0xb5ec0000)
libXau.so.6 =>/usr/lib/i386-linux-gnu/libXau.so.6 (0xb5ebb000)
libXdmcp.so.6 =>/usr/lib/i386-linux-gnu/libXdmcp.so.6 (0xb5eb4000)
cs@Ubuntu32:
textasciitilde /MeinProgramm/MeinProgramm Libs$

```

As you see all library have been found and their load address is printed behind the na,e. If a library is missing, you usually see the address missing there or being zero.

46.0.46 How can I validate an email address?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can try this code:

Example:

```

Var re As RegEx
re = New RegEx
Var rm As RegExMatch

```

```

re.SearchPattern = " [a-z0-9!#$%&'*/+=?^_`' { | }
textasciitilde - ]+(?:\. [a-z0-9!#$%&'*/+=?^_`' { | }

```

```
textasciitilde - ] +)*@(?: [ a-z0-9 ] (?: [ a-z0-9- ] * [ a-z0-9 ] )?\.)+ [ a-z0-9 ] (?: [ a-z0-9- ] * [ a-z0-9 ] )?)"
rm = re.Search(editField1.Text)
```

```
if rm = Nil Then
StaticText2.text = editField1.Text + " not valid email"
Else
StaticText2.Text = editField1.Text + " is valid"
End if
```

Notes: Adapted from:
<http://www.regular-expressions.info/email.html>

46.0.47 How do I decode correctly an email subject?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following code can be used to decode an email subject including several encodings including Base 64.

Example:

```
Var src as string // input

Var theRegex as Regex
Var theRegexMatch as RegexMatch
Var result, infoCharset, encodedPart as string
Var theStart as Integer

if instr(src, "=?") >0 then
theRegex = new Regex
theRegex.Options.Greedy = false
theRegex.searchPattern = "(.*)=\?(.+)\?(Q | B)\?(.+)\?="
theRegexMatch = theRegex.search(src)
while theRegexMatch <>nil
theStart = theRegexMatch.subExpressionStartB(0) + len(theRegexMatch.subExpressionString(0))

result = result + theRegexMatch.subExpressionString(1)
infoCharset = theRegexMatch.subExpressionString(2)
encodedPart = theRegexMatch.subExpressionString(4)
if theRegexMatch.subExpressionString(3) = "B" then
encodedPart = DecodeBase64(encodedPart)
elseif theRegexMatch.subExpressionString(3) = "Q" then
encodedPart = DecodeQuotedPrintable(encodedPart)
end if
if right(result, 1) = " " then
result = mid(result, 1, len(result)-1)
```

```

end if
encodedPart = encodedPart.DefineEncoding(GetInternetTextEncoding(infoCharset))
result = result + encodedPart

theRegex.SearchStartPosition = theStart
theRegexMatch = theRegex.search()
wend

result = result + mid(src, theStart+1)

else
result = src
end if
// theRegexMatch = theRegex.search

msgbox result

```

Notes: May not look nice depending on the controls used.
This is no longer needed when using MimeEmailMBS class which decodes for you.

46.0.48 How do I enable/disable a single tab in a tabpanel?

Plugin Version: all, Platform: macOS.

Answer: Use the TabpanelEnabledMBS method.

Example:

```
TabpanelEnabledMBS(tabpanel1, 1, false)
```

Notes: Use Carbon for MachO and CarbonLib for Mac Carbon and AppearanceLib for Mac OS Classic as library.

For Cocoa, please use enabled property of NSTabViewItemMBS class.

46.0.49 How do I find the root volume for a file?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this function:

Example:

```

Function GetRootVolume(f as FolderItem) as FolderItem
Var root, dum as folderItem
if f <> nil then
root = f // f might be the volume

```

```

do
dum = root.parent
if dum <>nil then
root = dum
end if
loop until dum = nil
return root
end if
End Function

```

46.0.50 How do I get the current languages list?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

Var p as new CFPREFERENCESMBS
Var a as CFArrayMBS
Var s as CFStringMBS
Var o as CFOBJECTMBS
Var sa(-1) as string

o=p.CopyAppValue("AppleLanguages",".GlobalPreferences")

if o<>Nil then
a=CFArrayMBS(o)

Var i,c as Integer

c=a.Count-1
for i=0 to c
o=a.Item(i)

if o isa CFStringMBS then
s=CFStringMBS(o)
sa.Append s.str
end if
next
end if

MsgBox Join(sa,EndOfLine)

```

Notes: On Mac OS X you can get the list of current languages like this list:

de
en
ja
fr
es
it
pt
pt-PT
nl
sv
nb
da
fi
ru
pl
zh-Hans
zh-Hant
ko

Which has German (de) on the top for a German user.
This code has been tested on Mac OS X 10.5 only.

46.0.51 How do I get the Mac OS Version?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
Var i as Integer
if system.gestalt("sysv", i) then
//do this in an 'If' in case you don't get any value back at all and system.gestalt returns boolean
if i = &h750 then //If OS is 7.5
//do stuff
elseif i = &h761 then //If OS is 7.6.1
//do stuff
end if
end if
```

Notes: The MBS Plugin has a function `SystemInformationMBS.OSVersionString` for this.

46.0.52 How do I get the printer name?

Plugin Version: all.

Answer: For Mac OS Classic see the code below and for Mac OS X use the Carbon Print Manager Classes from the MBS Plugin.

Example:

```
Var s as String
Var i as Integer
```

```
s=app.ResourceFork.GetResource("STR",-8192)
if s<>"" then
i=ascb(leftb(s,1))
s=mid(s,2,i)
```

```
MsgBox s
end if
```

Notes: A note from Craig Hoyt:

After looking at your example I had a little deja-vu experience. Several years ago I played around with this same code in FutureBasic. I discovered that it did not and still doesn't provide the 'Printer Name', it does return the print driver name. If it returns 'LaserWriter 8' as the print driver you can look into this file and get the 'PAPA' resource #-8192 to get the actual Printer Name. Unfortunately this does not hold true for other printers. My Epson and HP Printers (the Epson has an Ethernet Card and the HP is USB) do not provide this info in their drivers. As far as I can tell it only returns the name by polling the printer itself.

46.0.53 How do I make a metal window if RB does not allow me this?

Plugin Version: all, Platform: macOS.

Answer: The following declare turns any window on Mac OS X 10.2 or newer into a metal one.

Example:

```
declare sub ChangeWindowAttributes lib "Carbon" (win as windowptr, a as Integer, b as Integer)

ChangeWindowAttributes window1,256,0
```

Notes: May not look nice depending on the controls used.

46.0.54 How do I make a smooth color transition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

I'd like to show in a report some bars, which start with color A and end with color B.

The color change should be very smooth.

My problem: If I would start from 255,0,0 and end by 0,0,0, I would have 255 different colors. If the bars are longer than 255 pixels, would this look nice?

Example:

```
// Window.Paint:
Sub Paint(g As Graphics)
  Var w,w1,x,p as Integer
  Var c1,c2,c as color
  Var p1,p2 as Double

  c1=rgb(255,0,0) // start color
  c2=rgb(0,255,0) // end color

  w=g.Width
  w1=w-1

  for x=0 to w1
    p1=x/w1
    p2=1.0-p1

    c=rgb(c1.red*p1+c2.red*p2, c1.green*p1+c2.green*p2, c1.blue*p1+c2.blue*p2)

    g.ForeColor=c
    g.DrawLine x,0,x,g.Height

  next
End Sub
```

Notes:

Try the code above in a window paint event handler.

46.0.55 How do I read the applications in the dock app?

Plugin Version: all, Platform: macOS.

Answer: Use CFPREFERENCESMBS class like in this example:

Example:

```
// Reads file names from persistent dock applications and puts them into the list

Var pref as new CFPREFERENCESMBS

Var persistentapps as CFStringMBS = NewCFStringMBS("persistent-apps")
Var ApplicationID as CFStringMBS = NewCFStringMBS("com.apple.dock")
Var tiledata as CFStringMBS = NewCFStringMBS("tile-data")
Var filelabel as CFStringMBS = NewCFStringMBS("file-label")

// get the array of persistent applications from dock preferences
Var o as CFOBJECTMBS = pref.CopyValue(persistentapps, ApplicationID, pref.kCFPreferencesCurrentUser,
pref.kCFPreferencesAnyHost)

if o isa CFArrayMBS then
  Var a as CFArrayMBS = CFArrayMBS(o)

  // walk over all items in array
  Var c as Integer = a.Count-1
  for i as Integer = 0 to c

    // get dictionary describing item
    o = a.Item(i)

    if o isa CFDictionaryMBS then
      Var d as CFDictionaryMBS = CFDictionaryMBS(o)

      // and pick tile data dictionary
      o = d.Value(tiledata)
      if o isa CFDictionaryMBS then
        d = CFDictionaryMBS(o)

        // and pick there the file label
        o = d.Value(filelabel)
        if o isa CFStringMBS then
          // and display it
          Var name as string = CFStringMBS(o).str
          List.AddRow name
        end if
      end if
    end if
  next
end if
```

```
else
MsgBox "Failed to read dock preferences."
end if
```

Notes: You can use the `CFPreferencesMBS.SetValue` to change a value and `CFPreferencesMBS.Synchronize` to write the values to disc. You may need to restart the `Dock.app` if you modified things.

46.0.56 How do I truncate a file?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In a `binarystream` you can set the `length` property to truncate.

46.0.57 How do update a Finder's windows after changing some files?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
Var f as folderitem // some file
Var ae as appleevent
ae=newappleevent("fndr","fupd","MACS")
ae.folderitemparam("—")=f
if not ae.send then
//something went wrong
end if
```

Notes: The `folderitem.finderupdate` from the MBS Plugin does something like this.

46.0.58 How to access a USB device directly?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: First, it depends on the device.

Notes: Some devices can be talked directly from user mode code, but some require a kernel driver.

For some devices you can use plugins to access them like:

- Audio and Video sources using the `QTGrabberClassMBS`

- Mass storage devices using the folderitem class.
- Serial devices using the System.SerialPort function.
- HID USB devices can be used with MacHIDMBS, WinHIDMBS or LinuxHIDInterface class.
- Any USB device may be used with MacUSBMBS or WinUSBMBS classes.

In general it is always the best to take the most high level access to have others do the work for the details.

46.0.59 How to add icon to file on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use FolderItem.AddCustomIcon or NSWorkspaceMBS.setIcon functions.

Notes: Please close any open stream for the file you want to add an icon.

46.0.60 How to ask the Mac for the Name of the Machine?

Plugin Version: all, Platform: macOS.

Answer: Using Apple Events you can use this code:

Example:

Function Computername() *As string*

Var theEvent *as* AppleEvent

Var err *as* boolean

theEvent = newAppleEvent("mchn","getd","MACS")

err = theEvent.send

return theevent.ReplyString

End Function

Notes: Code above is for Mac OS 9!

Also the MBS Plugin has a function for this which may be faster and work also on Macs without Filesharing (which handles this event).

46.0.61 How to automatically enable retina in my apps?

Plugin Version: all, Platform: macOS.

Answer: You can run a build script on each build with this code:

Example:

```
Var App As String = CurrentBuildLocation + "/" + CurrentBuildAppName + ".app"
Call DoShellCommand("/usr/bin/defaults write " + App + "/Contents/Info ""NSHighResolutionCapable""
YES")
```

Notes: This will set the NSHighResolutionCapable flag to YES.

46.0.62 How to avoid leaks with Cocoa functions?

Plugin Version: all, Platform: macOS.

Answer: You can try this code on Mac OS X:

Example:

```
// in a Timer Action event:
Sub Action()
static LastPool as NSAutoreleasePoolMBS = nil
static CurrentPool as NSAutoreleasePoolMBS = nil

LastPool = CurrentPool
CurrentPool = new NSAutoreleasePoolMBS
End Sub
```

Notes: With Xojo 2009r4 the code above should not be needed as Xojo runtime does automatically handle the NSAutoreleasePools for you. For older Xojo versions you need to use code with a timer with the action event above to avoid memory leaks.

Please do not use Xojo 2009r4 and newer with plugins before version 9.5. You can get crashes there which typically show a line with a objc_msgSend call.

46.0.63 How to avoid trouble connecting to oracle database with SQL Plugin?

Plugin Version: all, Platform: macOS.

Answer: For oracle the most important thing is to point the plugin to the libraries from oracle.

Notes: In environment variables, the paths like ORACLE_HOME must be defined.

On Mac OS X you also need to define DYLD_LIBRARY_PATH to point to the dylib files from oracle.

For that you need to modify /etc/launchd.conf for Mac OS X 10.8 and newer.

In older versions those variables in .MacOSX/environment.plist file in user's home.

Another way for the case you bundle things inside your app is to use the LSEnvironment key in info.plist. In info.plist it looks like this:

```
<key>LSEnvironment</key>
<dict>
<key>test</key>
<string>Hello World</string>
</dict>
```

46.0.64 How to avoid ___NSAutoreleaseNoPool console messages in threads?

Plugin Version: all, Platform: macOS.

Answer: You need to use your own NSAutoreleasePool on a thread like this:

Example:

```
sub MyThread.run
Var pool as new NSAutoreleasePoolMBS
// do work here

pool=nil
end sub
```

Notes: For more details read here:

http://developer.apple.com/mac/library/documentation/Cocoa/Reference/Foundation/Classes/NSAutoreleasePool_Class/Reference/Reference.html

46.0.65 How to bring app to front?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac you can use this code:

Example:

```
// First way:
app.FrontMostMBS = true

// second way:
Var p as new ProcessMBS
p.GetCurrentProcess
p.FrontProcess = true

// third way:
NSApplicationMBS.sharedApplication.activateIgnoringOtherApps(true)
```

```
// for Windows:
RemoteControlMBS.WinBringWindowToTop
```

Notes: This will bring a Mac app to the front layer.

46.0.66 How to bring my application to front?

Plugin Version: all, Platform: macOS.

Answer: This makes SimpleText (Code ttxt) to the frontmost application:

Example:

```
Var A As AppleEvent
A = NewAppleEvent("misc", "actv", "")
If Not A.Send then
Beep
end if
```

Notes: (Code is Mac only)

46.0.67 How to catch Control-C on Mac or Linux in a console app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use SignalHandlerMBS class for this.

Example:

```
// watch for Control-C on Mac
call SignalHandlerMBS.SetFlagHandler(2)

Var ende as boolean = false
do
if SignalHandlerMBS.IsFlagSet(2) then
Print "Flag 2 set. Existing..."
ende = true
end if

DoEvents 1
loop until ende
```

Notes: The signal is caught, a flag is set and you can ask later in your normal application flow for the result.

46.0.68 How to change name of application menu?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Use this code to change the application menu name on Mac OS X:

Example:

```
Var mb as new MenubarMBS
Var m as MenuMBS = mb.item(1) // 1 is in my tests the app menu
if m<>Nil then
m.MenuTitle = "Hello World"
end if
```

Notes: This code is for Carbon only.

46.0.69 How to change the name in the menubar of my app on Mac OS X?

Plugin Version: all, Platform: macOS.

Answer:

You mean it screws up if the file name of the bundle itself is different than the name of the executable file in the MacOS folder within the bundle? If so, you should find something like this within your Info.plist file (or the 'plst' resource that the RB IDE builds for you):

```
<key>CFBundleExecutable</key>
<string>Executable file name here</string>
```

Just make sure that file name matches.

However, if your question involves how you can change the name of the app that appears in the menu and the dock, that's different. You can make this name different from the file name by changing the CFBundleName key:

```
<key>CFBundleName</key>
<string>Name for menu here</string>
```

Note that if you use my free AppBundler program, this second part is taken care of for you – just fill in a custom name in the right field.

You can find AppBundler (from Thomas Reed) at <http://www.bitjuggler.com/products/appbundler/>.

46.0.70 How to check if a folder/directory has subfolders?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this to check all items in a folder:

Example:

```
Function HasSubFolder(folder as FolderItem) As Boolean
  Var c as Integer = folder.Count
```

```
  for i as Integer = 1 to c
    Var item as FolderItem = folder.TrueItem(i)
```

```
    if item<>Nil and item.Directory then
      Return true
    end if
  next
```

```
End Function
```

Notes: We use trueitem() here to avoid resolving alias/link files.
Also we check for nil as we may not have permission to see all items.
And if one is a directory, we return without checking the rest.

46.0.71 How to check if Macbook runs on battery or AC power?

Plugin Version: all, Platform: macOS.

Answer: Please use our IOPowerSourcesMBS class like this:

Example:

```
Function PowerSourceState() as Integer
  Var p as new IOPowerSourcesMBS
```

```
  // check all power sources
  Var u as Integer = p.Count-1
  for i as Integer = 0 to u
    Var d as CFDictionaryMBS = p.Item(i)
    if d<>nil then
      // check if they have a power source state key:
      Var o as CFObjectMBS = d.Value(NewCFStringMBS("Power Source State"))
      if o isa CFStringMBS then
        Var s as string = CFStringMBS(o).str
```

```
      MsgBox s
```

```
      if s = "AC Power" then
```



```

Return 1
elseif s = "Battery Power" then
Return 2
end if
end if
end if
next
Return 0 // unknown
End Function

```

Notes: If you want to check the CFDictionaryMBS content, simply use a line like "dim x as dictionary = d.dictionary" and check the contents in the debugger.

46.0.72 How to check if Microsoft Outlook is installed?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you need Outlook for Scripting, you should simply check registry for the required Outlook.Application class:

Example:

```

Function OutlookInstalled() As Boolean
#if TargetWin32 then

try
Var r as new RegistryItem("HKEY_CLASSES_ROOT\Outlook.Application\CLSID", false)

Return true

catch r as RegistryAccessErrorException
// not installed
Return false

end try

#else

// Windows only, so false on other platforms
Return false

#endif

End Function

```

46.0.73 How to check on Mac OS which country or language is currently selected?

Plugin Version: all, Platform: macOS.

Answer: The code below returns a country value.

Example:

Var result as Integer

IF TargetMacOS THEN

CONST smScriptLang = 28

CONST smSystemScript = -1

DECLARE FUNCTION GetScriptManagerVariable LIB "Carbon" (selector as Integer) as Integer

DECLARE FUNCTION GetScriptVariable LIB "Carbon" (script as Integer, selector as Integer) as Integer

result=GetScriptVariable(smSystemScript, smScriptLang)

END IF

Notes: Returns values like:

For more values, check "Script.h" in the frameworks.

46.0.74 How to code sign my app with plugins?

Plugin Version: all, Platform: macOS.

Answer: When you try to code sign the application with plugin dylibs on Mac OS X, you may see error message that there is actually a signature included.

Notes: Please use the -f command line parameter with codesign utility to overwrite our MBS signature.

We sign our plugins for MacOS, iOS and Windows to make sure they have not been modified.

In terminal, you do like this:

```
cd <Path to folder of app>
```

```
xattr -cr <Appname>.app
```

```
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app/Contents/Frameworks/*.dylib
```

```
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app/Contents/Frameworks/*.framework
```

```
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app
```

Please use the name of your certificate (See keychain), the name of your app and the path to the app folder.
 If you have helper apps you need to sign them first.
 You can use a build step to automatically sign your app on build.

46.0.75 How to collapse a window?

Plugin Version: all, Platform: macOS.

Answer: Use this function (Mac only):

Example:

```
Sub CollapseRBwindow(w as window, CollapseStatus as boolean)
  Var state, err as Integer
  Var wh as MemoryBlock
```

```
Declare Function CollapseWindow Lib "Carbon" (window as Integer, collapse as Integer) as Integer
```

```
IF CollapseStatus THEN
  state = 1
ELSE
  state = 0
END IF
```

```
err = CollapseWindow(w.MacWindowPtr, state)
```

```
End Sub
```

Notes: Also the MBS Plugin has a window.collapsedmbs property you can set.
 For Windows the MBS Plugin has a window.isiconicmbs property.

46.0.76 How to compare two pictures?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can try this code:

Example:

```
Function ComparePictures(p as picture, q as picture) as Integer
  Var r, u as RGBSurface
  Var x, y, n, m, h, w as Integer
  Var w1, w2, h1, h2, d1, d2 as Integer
  Var c1, c2 as color
```

```
h1=p.Height
h2=q.Height
w1=p.Width
w2=q.Width
d1=p.Depth
d2=q.Depth

if d1<>d2 then
Return 1
elseif w1<>w2 then
return 2
elseif h1<>h2 then
Return 3
else
r=p.RGBSurface
u=q.RGBSurface

if r=nil or u=nil then
Return -1
else
h=h1-1
w=w1-1
m=min(w,h)

for n=0 to m
c1=r.Pixel(n,n)
c2=u.Pixel(n,n)
if c1<>c2 then
Return 4
end if
next

for y=0 to h
for x=0 to w
c1=r.Pixel(x,y)
c2=u.Pixel(x,y)
if c1<>c2 then
Return 5
end if
next
next

// 0 for equal
// -1 for error (no RGBsurface)
// 1 for different depth
// 2 for different width
// 3 for different height
// 4 for different pixels (fast test)
```

```
// 5 for different pixels (slow test)
end if
end if
```

```
Exception
Return -1
End Function
```

Notes: Remember that this only works on bitmap pictures, so the picture.BitmapMBS function may be useful.

46.0.77 How to compile PHP library?

Plugin Version: all, Platform: macOS.

Answer: You have to download the source code and compile a static version of the library.

Notes: This instructions were written based on PHP 5.2.6 on Mac OS X:

- Best take a new Mac with current Xcode version installed.
- Download the source code archive. e.g. "php-5.2.6.tar.bz2"
- Expand that archive on your harddisc.
- Open terminal window
- change directory to the php directory. e.g. "cd /php-5.2.6"
- execute this two lines to define the supported CPU types and the minimum Mac OS X version:
- export CFLAGS="-arch ppc -arch i386 -mmacosx-version-min=10.3"
- export CXXFLAGS="-arch ppc -arch i386 -mmacosx-version-min=10.3"
- the command "./configure help" does show the configure options.
- use configure with a line like this:
- ./configure --enable-embed --with-curl --enable-ftp --enable-zip --enable-sockets --enable-static --enable-soap --with-zlib --with-bz2 --enable-exif --enable-bcmath --enable-calendar
- start the compilation with "make all"
- other option is to use "make install" which first does the same as "make all" and than does some installation scripts.
- you may get an error about a duplicate symbole __yytext. Search the file "zend_ini_scanner.c", search a line with "char *yytext;" and change it to "extern char *yytext;"

- On the end you get a lot of error messages, but you have a working library (named libphp5.so) file in the invisible ".libs" folder inside your php source folder.

Possible problems and solutions:

- If the path to your files has spaces, you can get into trouble. e.g. "/RB Plugins/PHP" is bad as files will be searched sometimes in "/RB".
- If you have in /usr/local/lib libraries which conflict with the default libraries, you can get into trouble.
- If you installed some open source tools which compiled their own libraries, you can get into conflicts.
- if you have to reconfigure or after a problem, you may need to use "make clean" before you start "make all" again.

Feel free to install additional libraries and add more packages to the configure line.

46.0.78 How to convert a `BrowserType` to a `String` with `WebSession.Browser`?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetBrowserName(s as WebSession.BrowserType) As string
Select case s
case WebSession.BrowserType.Android
Return "Andriod"
case WebSession.BrowserType.Blackberry
Return "Blackberry"
case WebSession.BrowserType.Chrome
Return "Chrome"
case WebSession.BrowserType.ChromeOS
Return "ChromeOS"
case WebSession.BrowserType.Firefox
Return "Firefox"
case WebSession.BrowserType.InternetExplorer
Return "InternetExplorer"
case WebSession.BrowserType.Opera
Return "Opera"
case WebSession.BrowserType.Safari
Return "Safari"
case WebSession.BrowserType.SafariMobile
Return "SafariMobile"
case WebSession.BrowserType.Unknown
Return "Unknown"
```

```

else
Return "Unkown: "+str(integer(s))
end Select

```

End Function

46.0.79 How to convert a EngineType to a String with WebSession.Engine?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```

Function GetRenderingEngineName(s as WebSession.EngineType) As string
Select case s
case WebSession.EngineType.Gecko
Return "Gecko"
case WebSession.EngineType.Presto
Return "Presto"
case WebSession.EngineType.Trident
Return "Trident"
case WebSession.EngineType.Unknown
Return "Unknown"
case WebSession.EngineType.WebKit
Return "WebKit"
else
Return "Unkown: "+str(integer(s))
end Select

```

End Function

46.0.80 How to convert a PlatformType to a String with WebSession.Platform?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```

Function GetPlatformName(s as WebSession.PlatformType) As string
Select case s
case WebSession.PlatformType.Blackberry
Return "Blackberry"
case WebSession.PlatformType.iPad
Return "iPad"
case WebSession.PlatformType.iPhone

```

```

Return "iPhone"
case WebSession.PlatformType.iPodTouch
Return "iPodTouch"
case WebSession.PlatformType.Linux
Return "Linux"
case WebSession.PlatformType.Macintosh
Return "Macintosh"
case WebSession.PlatformType.PS3
Return "PS3"
case WebSession.PlatformType.Unknown
Return "Unknown"
case WebSession.PlatformType.WebOS
Return "WebOS"
case WebSession.PlatformType.Wii
Return "Wii"
case WebSession.PlatformType.Windows
Return "Windows"
else
Return "Unkown: "+str(integer(s))
end Select

```

End Function

46.0.81 How to convert a text to iso-8859-1 using the TextEncoder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

This code can help you although it's not perfect.

You need to set lc to the current color you use.

Example:

```

Var outstring as string
Var theMac, thePC as textencoding
Var Mac2PC as textconverter

theMac = getTextEncoding(0) // MacRoman
thePC = getTextEncoding(&h0201) // ISOLatin1

Mac2PC = getTextConverter(theMac, thePC)
// if you wanted to do the opposite just create a converter
// PC2Mac = getTextConverter(thePC, theMac)

outstring = Mac2PC.convert("Bj√rn, this text should be converted")
Mac2PC.clear

```


Notes:

You have to call `Mac2PC.clear` after every conversion to reset the encoding engine. See also newer `TextConverterMBS` class.

46.0.82 How to convert ChartTime back to Xojo date?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have this example code:

Example:

```
Function ChartTimeToDate(ChartTime as Double) As date
static diff as Double = 0.0
```

```
if diff = 0.0 then
Var d2 as Double = CDBaseChartMBS.chartTime(2015, 1, 1)
Var da as new date(2015, 1, 1)
Var ts as Double = da.TotalSeconds
```

```
diff = ts - d2
end if
```

```
Var d as new date
d.TotalSeconds = diff + ChartTime
```

```
Return d
End Function
```

Notes: As you see we calculate the difference in base date from `Date` and `ChartTime` and later use difference to convert.

46.0.83 How to convert line endings in text files?

Plugin Version: all, Platform: macOS.

Answer: You can simply read file with `TextInputStream` and write with new line endings using `TextOutputStream` class.

Example:

```
Var inputfile as FolderItem = SpecialFolder.Desktop.Child("test.txt")
Var outputfile as FolderItem = SpecialFolder.Desktop.Child("output.txt")
Var it as TextInputStream = TextInputStream.Open(inputfile)
Var ot as TextOutputStream = TextOutputStream.Create(outputfile)
```

```

ot.Delimiter = EndOfLine.Windows // new line ending
while not it.EOF
ot.WriteLine it.ReadLine
wend

```

Notes: TextInputStream will read any input line endings and with delimiter property in TextOutputStream you can easily define your new delimiter.

46.0.84 How to convert picture to string and back?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use this plugin functions:

Notes: JPEG:

```

JPEGStringToPictureMBS(buf as string) as picture
JPEGStringToPictureMBS(buf as string,allowdamaged as Boolean) as picture
PictureToJPEGStringMBS(pic as picture,quality as Integer) as string

```

PNG:

```

PictureToPNGStringMBS(pic as picture, gamma as single) as string
PictureToPNGStringMBS(pic as picture, mask as picture, gamma as single) as string
PictureToPNGStringMBS(pic as picture, gamma as single, Interlace as Boolean, FilterType as Integer) as string
PictureToPNGStringMBS(pic as picture, mask as picture, gamma as single, Interlace as Boolean, FilterType as Integer) as string
PNGStringToPictureMBS(data as string, gamma as single) as picture
PNGStringToPNGPictureMBS(data as string, gamma as single) as PNGpictureMBS

```

Tiff:

```

TIFFStringToPictureMBS(data as string) as picture
TIFFStringToTiffPictureMBS(data as string) as TiffPictureMBS

```

BMP:

```

BMPStringtoPictureMBS(data as string) as picture
Picture.BMPDataMBS(ResolutionValueDPI as Integer=72) as string

```

GIF:

```
GifStringToGifMBS(data as string) as GIFMBS
GifStringToPictureMBS(data as string) as Picture
```

46.0.85 How to copy an array?

Plugin Version: all, Platform: macOS.

Answer: You can use a function like this to copy an array:

Example:

```
Function CopyArray(a() as Double) as Double()
Var r() as Double
for each v as Double in a
r.Append v
next
Return r
End Function
```

Notes: If needed make several copies of this method with different data types, not just double. For a deep copy of an array of objects, you need to change code to also make a copy of those objects.

46.0.86 How to copy a dictionary?

Plugin Version: all, Platform: macOS.

Answer: You can use a function like this to copy a dictionary:

Example:

```
Function CopyDictionary(d as Dictionary) As Dictionary
Var r as new Dictionary
for each key as Variant in d.keys
r.Value(key) = d.Value(key)
next
Return r
End Function
```

Notes: If needed make several copies of this method with different data types, not just double. For a deep copy of a dictionary of objects, you need to change code to also make a copy of those objects.

46.0.87 How to copy parts of a movie to another one?

Plugin Version: all, Platforms: macOS, Windows.

Answer: The code below copies ten seconds of the snowman movie to the dummy movie starting at the 5th second.

Example:

```

Var f as FolderItem
Var md as EditableMovie
Var ms as EditableMovie

f=SpecialFolder.Desktop.Child("Our First Snowman.mov")
ms=f.OpenEditableMovie

ms.SelectionStartMBS=5
ms.SelectionLengthMBS=10

f=SpecialFolder.Desktop.Child("dummy.mov")
md=f.CreateMovie

msgbox str(md.AddMovieSelectionMBS(ms))

```

Notes: If result is not 0, the method fails.

46.0.88 How to create a birthday like calendar event?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

// start a connection to the calendar database
Var s as new CalCalendarStoreMBS

// needed for the error details
Var e as NSErrorMBS

Var r as CalRecurrenceRuleMBS = CalRecurrenceRuleMBS.initYearlyRecurrence(1, nil) // repeat every
year without end

Var a as new CalAlarmMBS // add alarm
a.action = a.CalAlarmActionDisplay
a.relativeTrigger = -3600*24 // 24 Hours before

// create a new calendar
Var c as new CalEventMBS

```

```

Var d as new date(2011, 04, 20) // the date

Var calendars() as CalCalendarMBS = s.calendars

// set properties
c.Title="Test Birthday"
c.startDate=d
c.recurrenceRule = r
c.calendar=calendars(0) // add to first calendar
c.addAlarm(a)
c.endDate = d
c.isAllDay = true

// save event
call s.saveEvent(c,s.CalSpanAllEvents, e)
if e<>nil then
MsgBox e.localizedDescription
else
MsgBox "New event was created."
end if

```

Notes: This adds an event to iCal for the given date with alarm to remember you and repeats it every year.

46.0.89 How to create a GUID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the UUIDMBS class for this.

46.0.90 How to create a Mac picture clip file?

Plugin Version: all, Platform: Windows.

Answer: You can use code like this one.

Example:

```

Var f As FolderItem
Var p As Picture

f=SpecialFolder.Desktop.Child("Test.pictClipping")
if f=nil then Return

```

```

p=new Picture(300,200,32) 'Make a sample picture
p.Graphics.ForeColor=RGB(0,255,255)
p.Graphics.FillOval 0,0,99,99
p.Graphics.ForeColor=RGB(255,0,0)
p.Graphics.DrawOval 0,0,99,99

```

Var r As ResourceFork 'ResourceFork is needed for a clip file

```

// Please define a file type Any
r=f.CreateResourceFork("Any")

```

```

// get PICT data using plugin function
Var pictdata as string = p.PicHandleDataMBS
r.AddResource(pictdata,"PICT",256,"Picture")

```

Var m as new MemoryBlock(8)

```

m.LittleEndian = false
m.Int16Value(0) = 0
m.Int16Value(2) = 0
m.Int16Value(4) = p.Width
m.Int16Value(6) = p.Height

```

```

r.AddResource(m,"RECT",256,"")

```

'Values taken from a sample file and irrelevant to the problem

```

Var data as string = DecodeBase64("AQAAAAAAAAAAAAAAAAACAFRDRVIAAAAABAAAAAAAAAAAAAAAAABUQ0IQAAAAA")
r.AddResource(data,"drag",128,"") 'ditto
r.Close

```

Notes: In general Apple has deprecated this, but a few application still support clippings.

46.0.91 How to create a PDF file in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Check our DynaPDF plugin and the examples.

Notes: An alternative can be to use the CoreGraphics and Cocoa functions on Mac OS X.

For Windows, we can only suggest our DynaPDF plugin.

46.0.92 How to create EmailAttachment for PDF Data in memory?

Plugin Version: all, Platform: macOS.

Answer: You can use code like the one below:

Example:

```
Function EmailAttachmentFromPDFData(PDFData as string, filename as string) As EmailAttachment
Var a as new EmailAttachment
```

```
a.data = EncodeBase64(PDFData, 76)
a.ContentEncoding = "base64"
a.MIMETYPE = "application/pdf"
a.MacType = "PDF "
a.MacCreator = "prvw"
a.Name = filename
```

Return a

End Function

Notes: Compared to sample code from Xojo documentation, we set the mime type correct for PDF. The MacType/MacCreator codes are deprecated, but you can still include them for older Mac email clients. "prvw" is the creator code for Apple's preview app.

46.0.93 How to create PDF for image files?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use DynaPDF like this:

Example:

```
Function CreatePrintPDF(jpgFiles() as folderitem, pdfFile as FolderItem, PageWidth as Integer, PageHeight
as Integer) As Boolean
// have files?
If pdfFile = Nil Then Return False
If jpgFiles = Nil Then Return False

If jpgFiles.Ubound < 0 Then Return False

// new DynaPDF
Var pdf As New MyDynapdfMBS

// page width/height in MilliMeter
Var pdfWidth as Integer = PageWidth * 72 / 25.4
Var pdfHeight as Integer = PageHeight * 72 / 25.4

// put your license here
Call pdf.SetLicenseKey "Starter"

// create pdf
```

```

Call pdf.CreateNewPDF pdfFile

// set a couple of options
Call pdf.SetPageCoords(MyDynaPDFMBS.kpcTopDown)
Call pdf.SetResolution(300)
Call pdf.SetUseTransparency(False)
Call pdf.SetSaveNewImageFormat(False)
Call pdf.SetGStateFlags(MyDynaPDFMBS.kgfUseImageColorSpace, False)
Call pdf.SetJPEGQuality(100)

// set page size
Call pdf.SetBBox(MyDynaPDFMBS.kpbMediaBox, 0, 0, pdfWidth, pdfHeight)
Call pdf.SetPageWidth(pdfWidth)
Call pdf.SetPageHeight(pdfHeight)

// append pages with one image per page
For i as Integer = 0 To jpgFiles.Ubound
Call pdf.Append
Call pdf.InsertImageEx(0, 0, pdfWidth, pdfHeight, jpgFiles(i), 1)
Call pdf.EndPage
Next

// close
Call pdf.CloseFile

Return True
End Function

```

Notes: This is to join image files in paper size to a new PDF.
e.g. scans in A4 into an A4 PDF.

46.0.94 How to CURL Options translate to Plugin Calls?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Below a few tips on how to translate command line CURL calls to plugin calls.

Notes: `curl -vX PUT http://localhost:5984/appserials/78569238475/DocumentRegister.docx?rev=3-25634563456 -data-binary @DocumentRegister.docx -H "Content-Type: application/msword"`

- The option -v means verbose. You can use OptionVerbose and listen for messages in the DebugMessage event.
- The option -X PUT means we want to do a HTTP PUT Request. So set OptionPut to true. Also you will want to set OptionUpload to true as you upload data.
- We have the URL which you put into OptionURL property.

- The `-data-binary` option tells CURL to pass the given data. With the `@` before the data, it is interpreted as a file name, so the data is read from the given file. You'll need to open this file and pass data with the Read event as needed. (See CURLS ftp file upload example project)
- The last option `-H` specifies an additional header for the upload. Pass this additional header with the `SetOptionHTTPHeader` method.

```
curl -X PUT http://127.0.0.1:5984/appserials/f2f4e540bf8bb60f61cfd4328001c59 -d ' { "type": "Product", "description": "Application Serial", "acronym": "AppSerial", "dateAdded": "2011-03-21 14:57:36" } '
```

- Option `-X PUT` like above.
- Pass the URL again in `OptionURL`
- This time data is passed in command line for CURL. You'd put this data in the quotes into a string and make it available in the Read event. (See CURLS ftp upload example project)

46.0.95 How to delete file with ftp and curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can set post/pre quotes to have ftp commands executed before or after the download/upload.

Example:

```
Var d as CURLMBS // your curl object
```

```
// delete file
```

```
Var ws() As String
```

```
ws.Append "DELE Temp.txt"
```

```
d.SetOptionPostQuote(ws)
```

Notes: Use `SetOptionPostQuote`, `SetOptionPreQuote` or `SetOptionQuote`.

The ftp commands you pass here are native ftp commands and not the commands you use with ftp applications. To delete use `DELE` and the file path.

46.0.96 How to detect display resolution changed?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac OS X simply listen for display changed notifications.

Notes: Use the "Distribution Notification Center.rbp" example project as a base and use it to listen to notifications with the name "O3DeviceChanged".

46.0.97 How to detect retina?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use `Window.BackingScaleFactorMBS` to query the factor.

Example:

```
msgbox str(window1.BackingScaleFactorMBS)
```

46.0.98 How to disable force quit?

Plugin Version: all, Platform: macOS.

Answer:

Please visit this website and get the control panel for Mac OS 9 there:

<http://www3.sk.sympatico.ca/tinyjohn/DFQ.html>

For Mac OS X use the MBS Plugin with the `SetSystemUIModeMBS` method.

Notes:

Please use `presentationOptions` in `NSApplicationMBS` for Cocoa applications.

46.0.99 How to disable the error dialogs from Internet Explorer on javascript errors?

Plugin Version: all, Platform: Windows.

Answer: You can use this code in the `htmlviewer` open event:

Example:

```
if targetwin32 then
htmlviewer1.__ole.Content.value("Silent") = True
end if
```

Notes: This disables the error dialogs from Internet Explorer.

46.0.100 How to display a PDF file in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac OS X you can use `CoreGraphics` or `PDFKit` to display a PDF.

Notes: An alternative can be to load the PDF into a `htmlviewer` so the PDF plugin can display it.

On Windows you may need to use the Acrobat ActiveX control from Adobe or launch Acrobat Reader.

46.0.101 How to do a lottery in RB?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this function:

Example:

```

Sub Lotto(max as Integer,count as Integer,z() as Integer)
// Lotto count numbers of max put into the array z beginning at index 0
Var n(0) as Integer ' all the numbers
Var m as Integer ' the highest field in the current array
Var i,a,b,d as Integer ' working variables

'fill the array with the numbers
m=max-1
redim n(m)

for i=0 to m
n(i)=i+1
next

' unsort them by exchanging random ones
m=max*10
for i=1 to m
a=rnd*max
b=rnd*max

d=n(a)
n(a)=n(b)
n(b)=d
next

' get the first count to the dest array
m=count-1
redim z(m)
for i=0 to m
z(i)=n(i)
next

'sort the result
z.sort
End Sub

Sub Open()
// Test it

```



```

ox=x
oy=y

x=x+dx
y=y+dy

if b then
g.DrawLine ox,oy,x,y
end if

b=not b
wend

End Sub

```

Notes: It would be possible to add this to the plugin, but I think it's better if you do it in plain Xojo code, so it even works on Windows.

46.0.104 How to draw a nice antialiased line?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

This code can help you although it's not perfect.
You need to set lc to the current color you use.

Example:

```

Sub drawLine(xs as Integer, ys as Integer, xe as Integer, ye as Integer, face as RGBSurface, lineColor as
color)
Var intX, intY, count, n, xDiff, yDiff as Integer
Var v, v1, floatX, floatY, xx, yy, xStep, yStep as Double
Var c as color

const st=1.0

xDiff=xe-xs
yDiff=ye-ys
count=max(abs(xDiff), abs(yDiff))
xStep=xDiff/count
yStep=yDiff/count
xx=xs
yy=ys
for n=1 to count
intX=xx

```

```

intY=yy
floatX=xx-intX
floatY=yy-intY

v=(1-floatX)*(1-floatY)*st
v1=1-v
c=face.pixel(intX, intY)
face.pixel(intX, intY)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=floatX*(1-floatY)*st
v1=1-v
c=face.pixel(intX+1, intY)
face.pixel(intX+1, intY)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=(1-floatX)*floatY*st
v1=1-v
c=face.pixel(intX, intY+1)
face.pixel(intX, intY+1)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=floatX*floatY*st
v1=1-v
c=face.pixel(intX+1, intY+1)
face.pixel(intX+1, intY+1)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)

xx=xx+xStep
yy=yy+yStep
next

```

End Sub

Notes:

PS: st should be 1 and face should be a RGBSurface or a Graphics object.

46.0.105 How to dump java class interface?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In terminal you can use "javap -s <classname>" to display the class with the method names and parameters.

Notes: For example show ResultSet class: javap -s java.sql.ResultSet

46.0.106 How to duplicate a picture with mask or alpha channel?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this function:

Example:

```

Function Duplicate(extends p as Picture) As Picture
  #if RBVersion >= 2011.04 then
  if p.HasAlphaChannel then

    // create nw picture and copy content:
    Var q as new Picture(p.Width, p.Height)
    q.Graphics.DrawPicture p,0,0

  Return q

  end if
  #endif

  // create new picture
  Var q as new Picture(p.Width, p.Height, 32)

  // get mask
  Var oldMask as Picture = p.mask(false)
  if oldMask = nil then
    // no mask, so simple copy
    q.Graphics.DrawPicture p,0,0
    Return q
  end if

  // remove mask
  p.mask = nil

  // copy picture and mask
  q.Graphics.DrawPicture p, 0, 0
  q.mask.Graphics.DrawPicture oldMask,0,0

  // restore mask
  p.mask = oldmask

  Return q
End Function

```

Notes: Simply copy it to a module and call it like this: q = p.duplicate.

The code above works with old Xojo versions because of the #if even if your RS version does not support alpha channel pictures. This way it's future proof.

46.0.107 How to enable assistive devices?

Plugin Version: all, Platform: macOS.

Answer: You can use AppleScript code like below:

Notes: tell application "System Events"
activate

```
set UI elements enabled to true
```

```
return UI elements enabled
end tell
```

You can run this with AppleScriptMBS class.

46.0.108 How to encrypt a file with Blowfish?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
Var fi as FolderItem = SpecialFolder.Desktop.Child("test.xojo_binary_project")
Var fo as FolderItem = SpecialFolder.Desktop.Child("test.encrypted")
```

```
// read input
Var bi as BinaryStream = BinaryStream.Open(fi)
Var si as string = bi.Read(bi.Length)
bi.Close

// encrypt
Var so as string = BlowfishMBS.Encrypt("MyKey",si)

// write output
Var bo as BinaryStream = BinaryStream.Create(fo)
bo.Write so
bo.Close
```

Notes: Of course you can decrypt same way, just use Decrypt function and of course swap files.

46.0.109 How to extract text from HTML?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use both RemoveHTMLTagsMBS and DecodingFromHTMLMBS like this:

Example:


```

Var html as string = "<p><B>Gr&uuml;&szlig;e</B></P>"
Var htmltext as string = RemoveHTMLTagsMBS(html)
Var text as string = DecodingFromHTMLMBS(htmltext)

```

MsgBox text // shows: Gr√üë

Notes: You can use it together with RemoveHTMLTagsMBS to remove html tags. What you get will be the text without tags.

DecodingFromHTMLMBS turns HTML escapes back to unicode characters. Like ä to √§.

46.0.110 How to find empty folders in a folder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this code:

Example:

```

Var folder as folderitem // your folder

Var c as Integer = folder.count
for i as Integer = 1 to c
  Var item as folderitem = folder.trueitem(i)
  if item = nil then
    // ignore
  elseif item.directory then
    // folder
    if item.count = 0 then
      // found empty folder
    end if
  end if
end if
next

```

46.0.111 How to find iTunes on a Mac OS X machine fast?

Plugin Version: all, Platform: macOS.

Answer: You can try Launch Services.

Example:

```

Var f as FolderItem

f=LaunchServicesFindApplicationForInfoMBS("hook","com.apple.iTunes","iTunes.app")

MsgBox f.NativePath

```

46.0.112 How to find network interface for a socket by it's name?

Plugin Version: all, Platform: macOS.

Answer: You can use our plugin to build a lookup table.

Example:

```
Function FindNetworkInterface(name as string) As NetworkInterface
name = name.trim

if name.len = 0 then Return nil

// search by IP/MAC
Var u as Integer = System.NetworkInterfaceCount-1
for i as Integer = 0 to u
Var n as NetworkInterface = System.GetNetworkInterface(i)
if n.IPAddress = name or n.MACAddress = name then
Return n
end if
next

// use MBS Plugin to build a mapping
Var interfaces() as NetworkInterfaceMBS = NetworkInterfaceMBS.AllInterfaces
Var map as new Dictionary

for each n as NetworkInterfaceMBS in interfaces
Var IPv4s() as string = n.IPv4s
Var IPv6s() as string = n.IPv6s

for each IPv4 as string in IPv4s
map.Value(IPv4) = n.Name
next
for each IPv6 as string in IPv6s
map.Value(IPv6) = n.Name
next
if n.MAC<>"" then
map.Value(n.MAC) = n.Name
end if
next

// now search interfaces by name, IPv4 or IPv6
for i as Integer = 0 to u
Var n as NetworkInterface = System.GetNetworkInterface(i)
if map.Lookup(n.IPAddress, "") = name then
Return n
end if

if map.Lookup(n.MACAddress, "") = name then
Return n
```

```
end if
next
```

End Function

Notes: The code above uses a lookup table build using NetworkInterfaceMBS class to find the network interface by name.

46.0.113 How to find version of Microsoft Word?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
// find Word
Var f as FolderItem = LaunchServicesFindApplicationForInfoMBS("", "com.microsoft.Word", "")

// open bundle
Var c as new NSBundleMBS(f)

// read info
Var d as Dictionary = c.infoDictionary

// show version
MsgBox d.Lookup("CFBundleVersion", "")
```

Notes: Older versions of Word can be found with creator code "MSWD".

46.0.114 How to fix CURL error 60/53 on connecting to server?

Plugin Version: all, Platform: macOS.

Answer: You probably connect with SSL and you have no valid certificate.

Example:

```
Var d as new CURLSMBS

// Disable SSL verification
d.OptionSSLVerifyHost = 0 // don't verify server
d.OptionSSLVerifyPeer = 0 // don't proofs certificate is authentic

// With SSL Verification:
Var cacert as FolderItem = Getfolderitem("cacert.pem")
```

```
d.OptionCAInfo = cacert.NativePath
d.OptionSSLVerifyHost = 2 // verify server
d.OptionSSLVerifyPeer = 1 // proofs certificate is authentic
```

Notes: You can either use the code above to disable the SSL verification and have no security. Or you use the cacert file and enable the verification. Than you only get a connection if the server has a valid certificate.

see also:

<http://curl.haxx.se/ca/>

46.0.115 How to format double with n digits?

Plugin Version: all, Platform: macOS.

Answer: You can use the FormatMBS function for this.

Example:

```
Var d as Double = 123.4567890
listbox1.AddRow FormatMBS("%f", d)
listbox1.AddRow FormatMBS("%e", d)
listbox1.AddRow FormatMBS("%g", d)

listbox1.AddRow FormatMBS("%5.5f", d)
listbox1.AddRow FormatMBS("%5.5e", d)
listbox1.AddRow FormatMBS("%5.5g", d)

d = 0.000000123456
listbox1.AddRow FormatMBS("%f", d)
listbox1.AddRow FormatMBS("%e", d)
listbox1.AddRow FormatMBS("%g", d)

listbox1.AddRow FormatMBS("%5.5f", d)
listbox1.AddRow FormatMBS("%5.5e", d)
listbox1.AddRow FormatMBS("%5.5g", d)
```

Notes: see FormatMBS for details.

In general %f is normal style, %e is scientific and %g is whichever gives best result for given space.

46.0.116 How to get a time converted to user time zone in a web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the WebSession.GMTOffset property.

Example:

```
Sub Open()
// current date on server
Var d as new date
Var s as string = d.LongTime

// adjust to client GMT offset
d.GMTOffset = d.GMTOffset + Session.GMTOffset

Var t as string = D.LongTime

MsgBox s+EndOfLine+t
End Sub
```

46.0.117 How to get an handle to the frontmost window on Windows?

Plugin Version: all, Platform: Windows.

Answer: This function returns a handle for the frontmost window:

Example:

```
Function GetForegroundWindowHandle() as Integer
#if targetwin32 then
declare function GetForegroundWindow Lib "user32.dll" as Integer
Return GetForegroundWindow()
#endif
End Function
```

46.0.118 How to get CFAbsoluteTime from date?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Use code like this:

Example:

```
Var d as new date
Var t as CFTimeZoneMBS = SystemCFTimeZoneMBS
Var g as new CFGregorianCalendarMBS
g.Day = d.Day
g.Month = d.Month
g.Year = d.Year
g.Minute = d.Minute
g.Hour = d.Hour
g.Second = d.Second
```

```
Var at as CFAbsoluteTimeMBS = g.AbsoluteTime(t)
Var x as Double = at.Value
```

```
MsgBox str(x)
```

Notes: As you see we need a timezone and put the date values in a gregorian date record. Now we can query absolute time for the given timezone.

46.0.119 How to get client IP address on web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the WebSession.RemoteAddress property.

Example:

```
Sub Open()
Title = Session.RemoteAddress
End Sub
```

46.0.120 How to get fonts to load in charts on Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use the SetFontSearchPath method in the CDBaseChartMBS class to specify where your fonts are.

Example:

```
if TargetLinux then
CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype;/usr/share/fonts/truetype/msttcorefonts"
else
// on Mac and Windows we use system fonts.
end if

// also you can later switch default fonts:

Var Chart as CDBaseChartMBS // your chart

#If TargetARM And TargetLinux Then
// use specific fonts on Linux on Raspberry Pi
Call Chart.setDefaultFonts("/usr/share/fonts/truetype/piboto/PibotoLt-Regular.ttf", "/usr/share/fonts/truetype/piboto/Pi
#EndIf
```

Notes: On macOS, iOS and Windows, the fonts are loaded from the system's font folder.

e.g. if you use ubuntu, you can install the ttf-mscorefonts-installer package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

46.0.121 How to get fonts to load in DynaPDF on Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use the `AddFontSearchPath` method in the `DynaPDFMBS` class to specify where your fonts are.

Example:

```
Var d as new DynaPDFMBS
if TargetLinux then
call d.AddFontSearchPath "/usr/share/fonts/truetype", true
else
// on Mac and Windows we use system fonts.
end if
```

Notes: On Mac OS X and Windows, the fonts are loaded from the system's font folder.

e.g. if you use ubuntu, you can install the ttf-mscorefonts-installer package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

46.0.122 How to get GMT time and back?

Plugin Version: all, Platform: macOS.

Answer: You can use the `date` class and the `GMTOffset` property.

Example:

```
// now
Var d as new date

// now in GMT
Var e as new date
e.GMTOffset = 0

// show
MsgBox str(d.TotalSeconds, "0.0")+ " " +str(e.TotalSeconds, "0.0")

Var GMTTimeStamp as Double = e.TotalSeconds

// restore
```

```
Var f as new date
```

```
// add GMT offset here
f.TotalSeconds = GMTTimeStamp + f.GMTOffset*3600
// because here it's removed
f.GMTOffset = f.GMTOffset
```

```
MsgBox d.ShortTime+" ("&str(d.GMTOffset)&") "&str(d.TotalSeconds,"0.0")+EndOfLine+_
e.ShortTime+" ("&str(e.GMTOffset)&") "&str(e.TotalSeconds,"0.0")+EndOfLine+_
f.ShortTime+" ("&str(f.GMTOffset)&") "&str(f.TotalSeconds,"0.0")
```

Notes: It's sometimes a bit tricky with the date class as setting one property often changes the others.

46.0.123 How to get good crash reports?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Check this website from the webkit website:

Notes: <http://webkit.org/quality/crashlogs.html>

46.0.124 How to get list of all threads?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the runtime module like in this function:

Example:

```
Function Threads() As Thread()
#pragma DisableBackgroundTasks
Var t() as Thread

Var o as Runtime.ObjectIterator=Runtime.IterateObjects
While o.MoveNext
if o.Current isa Thread then
t.Append thread(o.current)
end if
Wend

Return t
End Function
```

Notes: This returns an array of all thread objects currently in memory.

The pragma is important here as it avoids thread switches which may cause a thread to be created or deleted.

46.0.125 How to get parameters from webpage URL in Xojo Web Edition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the Webpage.ParametersReceived event.

Example:

```
Sub ParametersReceived(Variables As Dictionary)
for each key as Variant in Variables.keys
MsgBox key+" ->" +Variables.Value(key)
next
End Sub
```

Notes: The text encodings of this strings is not defined in Xojo 2010r5. Please use DefineEncoding.

46.0.126 How to get the color for disabled textcolor?

Plugin Version: all, Platform: macOS.

Answer: Ask the appearance manager:

Example:

```
Function GetThemeTextColor(inColor as Integer, inDepth as Integer, inColorDev as Boolean) As Color
declare function GetThemeTextColor lib "Carbon" (inColor as Integer, inDepth as Integer, inColorDev as
Boolean, outColor as Ptr) as Integer

Var i as Integer
Var col as MemoryBlock

col = newMemoryBlock(6)

i = GetThemeTextColor(inColor, inDepth, inColorDev, col)

return RGB(col.UShort(0)\256, col.UShort(2)\256, col.UShort(4)\256)
End Function
```

Notes: The color for this is:

```
const kThemeTextColorDialogInactive = 2.
```

```
c = GetThemeTextColor(kThemeTextColorDialogInactive, Screen(0).Depth, true)
```

For Mac OS X you should use "CarbonLib" instead of "AppearanceLib" ...

46.0.127 How to get the current free stack space?

Plugin Version: all, Platform: macOS.

Answer: You can something like the code below:

Example:

```
Sub ShowStackSize()
  Var threadid as Integer
  Var size as Integer

  declare function GetCurrentThread lib "Carbon" (byref threadid as Integer) as short
  declare function ThreadCurrentStackSize lib "Carbon" (threadid as Integer, byref size as Integer) as short

  if GetCurrentThread(threadid)=0 then
  if 0=ThreadCurrentStackSize(threadid,size) then
  MsgBox str(size)
  end if
  end if
End Sub
```

Notes: For Mac OS 9, use "ThreadLib" instead of "CarbonLib". You can use #if if you like for that.

46.0.128 How to get the current timezone?

Plugin Version: all, Platforms: macOS, Windows.

Answer:

You can use the TimeZoneMBS class or the CFTimeZoneMBS class.

Or code like below:

Example:

```
Function GMTOffsetInMinutes() as Integer
  // Returns the offset of the current time to GMT in minutes.
  // supports Mac OS and Windows, but not Linux yet (let me know if
  // you have code for that, please)
  //
  // Note that the offset is not always an even multiple of 60, but
  // there are also half hour offsets, even one 5:45h offset

  // This version by Thomas Tempelmann (rb@tempel.org) on 25 Nov 2005
  // with a fix that should also make it work with future Intel Mac targets.
  //
  // Using code from various authors found on the RB NUG mailing list

  Var result, bias, dayLightbias as Integer
```

```

Var info as memoryBlock
Var offset as Integer

#if targetMacOS then

Declare Sub ReadLocation lib "Carbon" (location As ptr)

info = NewMemoryBlock(12)
ReadLocation info
if false then
// bad, because it does not work on Intel Macs:
'offset = info.short(9) * 256 + info.byte(11)
else
offset = BitwiseAnd (info.long(8), &hFFFFFF)
end

offset = info.short(9) * 256 + info.byte(11)
offset = offset \60
return offset

#endif

#if targetWin32 then

Declare Function GetTimeZoneInformation Lib "Kernel32" ( tzInfoPointer as Ptr ) as Integer
// returns one of
// TIME_ZONE_ID_UNKNOWN 0
// - Note: e.g. New Delhi (GMT+5:30) and Newfoundland (-3:30) return this value 0
// TIME_ZONE_ID_STANDARD 1
// TIME_ZONE_ID_DAYLIGHT 2

info = new MemoryBlock(172)
result = GetTimeZoneInformation(info)

bias = info.Long(0)
// note: the original code I found in the NUG archives used Long(84) and switched to Long(0)
// only for result=1 and result=2, but my tests found that Long(0) is also the right value for result=0

if result = 2 then
daylightBias = info.long(168)
end if
offset = - (bias + dayLightbias)
return offset

#endif

End Function

```

46.0.129 How to get the current window title?

Plugin Version: all, Platform: macOS.

Answer: The code below returns the current window title for the frontmost window on Mac OS X if Accessibility services are

Example:

```
Function CurrentWindowTitle() As string
// your application needs permissions for accessibility to make this work!

Var SystemWideElement,FocusedApplicationElement,FocusedWindowElement as AXUIElementMBS
Var FocusedApplication,FocusedWindow,Title as AXValueMBS
Var s as String
Var cs as CFStringMBS

SystemWideElement=AccessibilityMBS.SystemWideAXUIElement
if SystemWideElement<>nil then
FocusedApplication=SystemWideElement.AttributeValue(AccessibilityMBS.kAXFocusedApplicationAttribute)
if FocusedApplication.Type=AccessibilityMBS.kAXUIElementMBSTypeID then
FocusedApplicationElement=new AXUIElementMBS
FocusedApplicationElement.Handle=FocusedApplication.Handle
FocusedApplicationElement.RetainObject

FocusedWindow=FocusedApplicationElement.AttributeValue(AccessibilityMBS.kAXFocusedWindowAttribute)

if FocusedWindow<>nil and AccessibilityMBS.kAXUIElementMBSTypeID=FocusedWindow.Type then

FocusedWindowElement=new AXUIElementMBS
FocusedWindowElement.Handle=FocusedWindow.Handle
FocusedWindowElement.RetainObject

Title=FocusedWindowElement.AttributeValue(AccessibilityMBS.kAXTitleAttribute)
if Title<>nil and Title.Type=kCFStringMBSTypeID then
cs=new CFStringMBS
cs.handle=Title.Handle
cs.RetainObject
Return cs.str
end if
end if
end if
end if
End Function
```

46.0.130 How to get the cursor blink interval time?

Plugin Version: all, Platform: macOS.

Answer: On Mac OS you can use GetCaretTime from the toolbox.

Example:

```
declare function GetCaretTime lib "Carbon" () as Integer
```

```
MsgBox str(GetCaretTime())+" ticks"
```

Notes: 60 ticks make one second.

46.0.131 How to get the list of the current selected files in the Finder?

Plugin Version: all, Platform: macOS.

Answer:

Use the AppleScript like this one:

```
tell application "finder"
return selection
end tell
```

Which translates into this AppleEvent:

```
Process("Finder").SendAE "core,getd,'—':obj { form:prop, want:type(prop), seld:type(sele), from:'null'() }
"
```

and as Xojo code it looks like this:

Example:

```
Var ae as appleEvent
Var o1 as appleEventObjectSpecifier
Var f as folderItem
Var aList as appleEventdescList
Var i as Integer
Var dateiname as string

// setup the AppleEvent
o1=getpropertyObjectDescriptor( nil, "sele")
ae= newappleEvent("core", "getd", "MACS")
ae.objectSpecifierParam("—")=o1
```

```

// send it
if ae.send then
// got the list
alist=ae.replyDescList

// now show the list of filename into an editfield:

for i=1 to alist.count
f=alist.folderItem(i)

dateiname=f.name
// editfield1 with property "multiline=true"!
editfield1.text=editfield1.text + dateiname + chr(13)
next
end if

```

46.0.132 How to get the Mac OS system version?

Plugin Version: all, Platform: macOS.

Answer: The following code queries the value and displays the version number:

Example:

```

Var first as Integer
Var second as Integer
Var third as Integer
Var l as Integer

if System.Gestalt("sysv",l) then

Third=Bitwiseand(l,15)
second=Bitwiseand(l\16,15)
first=Bitwiseand(l\256,15)+10*Bitwiseand(l\256\16,15)
end if

if First>=10 then
msgbox "Mac OS X "+str(First)+" "+str(Second)+" "+str(third)
else
msgbox "Mac OS "+str(First)+" "+str(Second)+" "+str(third)
end if

```

46.0.133 How to get the Mac OS Version using System.Gestalt?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

Var s As String
Var b As Boolean
Var i, resp as Integer

// Systemversion
b = System.Gestalt("sysv", resp)
If b then
s = Hex(resp)
For i =Len(s)-1 DownTo 1
s=Left(s,i)+" "+Mid(s,i+1)
Next
MsgBox "Systemversion: Mac OS " + s
end if

```

Notes: The MBS Plugin has a SystemInformationMBS.OSVersionString function for this.

46.0.134 How to get the screensize excluding the task bar?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Notes: Use the Screen class with the available* properties.

46.0.135 How to get the size of the foremost window on Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Notes: Make yourself a class for the WindowRect with four properties:

```

Bottom as Integer
Left as Integer
Right as Integer
Top as Integer

```

Add the following method to your class:

```

Sub GetWindowRect(windowhandle as Integer)
Var err as Integer
Var mem as memoryBlock

```

```
#if targetwin32 then
Declare Function GetWindowRect Lib "user32.dll" (hwnd as Integer, ipRect As Ptr) as Integer
```

```
mem = newmemoryBlock(16)
err = GetWindowRect(windowhandle, mem)
Left = mem.long(0)
Top = mem.Long(4)
Right = mem.Long(8)
Bottom = mem.Long(12)
#endif
End Sub
```

Good to use for the MDI Master Window!

46.0.136 How to get the source code of a HTMLViewer?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
// for Windows:
```

```
msgbox HTMLViewer1.IEHTMLTextMBS
```

```
// for MacOS with WebKit 2.x:
```

```
msgbox HTMLViewer1.WKWebViewMBS.HTMLText
```

46.0.137 How to get Xojo apps running Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You need to install some require packages.

Notes: You need CUPS as well as GTK packages. On 64 bit systems also the ia32-libs package.

Please note that you need a x86 compatible Linux. So no PPC, Power, ARM or other CPUs.

46.0.138 How to handle really huge images with GraphicsMagick or ImageMagick?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sometimes it may be better to use an extra application to process images.

Notes: A typical 32 bit app made with Xojo can use around 1.8 GB on Windows and 3 GB on Mac OS X. Some images may be huge, so that processing them causes several copies of the image to be in memory. With a 500 MB image in memory, doing a scale or rotation may require a temp image. So with source, temp and dest images with each 500 MB plus your normal app memory usage, you may hit the limit of Windows with 1.8 GB.

In that case it may be worth running a tool like gm in the shell class. gm is the command line version of GraphicsMagick. There you can run the 64 bit version which is not limited in memory like your own application. Also you can monitor progress and keep your app responsive.

46.0.139 How to handle tab key for editable cells in listbox?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this function:

Example:

```
Function HandleTabInList(list as listbox, row as Integer, column as Integer, key as String) As Boolean
// Handle tab character in Listbox.CellKeyDown event
```

```
Select case asc(key)
case 9
if Keyboard.AsyncShiftKey then
// back

// look for column left
for i as Integer = column-1 downto 0
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
```

```
// not found, so look in row before
row = row - 1
if row >= 0 then
for i as Integer = list.ColumnCount-1 downto 0
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
```

```

end if
next
end if
else
// forward

// look for column right
for i as Integer = column+1 to list.ColumnCount-1
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next

// not found, so look in row below
row = row + 1
if row <list.ListCount then
for i as Integer = 0 to list.ColumnCount-1
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
end if
end if
end Select
End Function

```

Notes: You call it from CellKeyDown event like this:

```

EventHandler Function CellKeyDown(row as Integer, column as Integer, key as String) As Boolean
if HandleTabInList(me, row, column, key) then Return true
End EventHandler

```

As you see in the code, we handle tab and shift + tab for moving back and forward. Also we wrap to previous/next row if needed. Feel free to extend this to wrap from last to first row or create a new row for editing.

46.0.140 How to hard link MapKit framework?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Our MapKit classes weak link the framework. If you need hard linking it for the App Store, you can add this method to a class:

Example:

```

Sub ReferenceMapKit()
// just put this in window or app class

#if TargetMachO and Target64Bit then
Declare sub testing Lib "MapKit" Selector "test" (id as ptr)
testing(nil)
#endif

End Sub

```

Notes: No need to call the method.
Just having it in a window or app, will cause the compiler to hard link the framework.

46.0.141 How to have a PDF downloaded to the user in a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use a WebHTMLViewer control and load the PDF file with the PDF plugin from the browser.

Example:

```

Var CurrentFile as WebFile // a property of the WebPage

// define the PDF file
CurrentFile = new WebFile
CurrentFile.Filename = "test.pdf"
CurrentFile.MIMEType = "application/pdf"
CurrentFile.Data = "some pdf data" // MyDynaPDF.GetBuffer
CurrentFile.ForceDownload = true

// start the download
showurl(CurrentFile.url)

```

Notes: See our Create PDF example for the Xojo Web Edition.

46.0.142 How to hide all applications except mine?

Platform: macOS.

Answer: The code below will on Mac OS hide all applications except your one:

Example:

```

Var p as new ProcessMBS

```

```
p.GetFirstProcess
do
if not p.FrontProcess then
p.Visible=false
end if
loop until not p.GetNextProcess
```

46.0.143 How to hide script errors in HTMLViewer on Windows?

Plugin Version: all, Platform: Windows.

Answer: Set Internet Explorer to silent mode with code like this:

Example:

```
htmlviewer1._ole.Content.value("Silent") = True
```

Notes: Simply put this code in the open event of your htmlviewer control (using me instead of htmlviewer1).

46.0.144 How to hide the grid/background/border in ChartDirector?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you want to hide something in a chart, simply assign the kTransparent constant as color.

46.0.145 How to hide the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub HideCursor Lib "Carbon" () Inline68K("A852")
```

```
HideCursor
```

Notes: The MBS Plugin has this function and supports it on Windows, too.

46.0.146 How to insert image to NSTextView or TextArea?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With NSTextViewMBS you can use this code to insert file:

Example:

```
// insert a file to textview

Public Sub InsertFile(textview as NSTextViewMBS, f as FolderItem)
// read to file
Var b as BinaryStream = BinaryStream.Open(f)
Var s as string = b.Read(b.Length)

// build wrapper
Var fileWrapper as NSFileWrapperMBS = NSFileWrapperMBS.initRegularFileWithContents(s)
fileWrapper.preferredFilename = f.name

// make attachment
Var fileAttachment as new NSTextAttachmentMBS(fileWrapper)
Var attributedString as NSAttributedStringMBS = NSAttributedStringMBS.attributedStringWithAttachment(fileAttachment)

// add to a NSTextViewMBS
textview.insertText attributedString

End Sub
```

Notes: For TextArea you can query the underlying NSTextViewMBS object via TextArea.NSTextViewMBS method.

46.0.147 How to jump to an anchor in a htmlviewer?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You can use javascript to change the current window's location.

Example:

```
// load website
htmlviewer1.LoadURL "http://www.monkeybreadsoftware.net/addressbook-abpersonmbs.shtml"

// later jump to anchor named "16":

if TargetWin32 then
call HTMLViewer1.IERunJavaScriptMBS "window.location = ""#16""
end if
```

46.0.148 How to keep a movieplayer unclickable?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: To keep the user away from clicking on a playing Movie you can just drop a Canvas in front of the Movieplayer and take the clicks there.

Example:

```
Function Canvas1.MouseDown(X as Integer, Y as Integer) as boolean
return true // take it and do nothing
End Function
```

46.0.149 How to keep my web app from using 100% CPU time?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Linux and MacOS you can use renice command in the terminal. On Windows use the task manager to reduce priority.

Notes: If you launch your app with nohup on Linux or Mac OS X like this from the terminal or a script:

```
nohup /webapps/MyApp/MyApp &
```

you can simply have a second line saying this:

```
renice 20 $ !
```

which tells the system to lower priority to lowest value for the latest background process.

46.0.150 How to kill a process by name?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can kill a process (or application) by name if you loop over all the processes and kill the one you need.

Example:

```
Var p as new ProcessMBS
p.GetfirstProcess ' get first
do
if p.name = "TextEdit" then
call p.KillProcess
Return
end if
```

```
loop until not p.GetNextProcess
```

Notes: You may want to check the result of killProcess function. Not every user is allowed to kill every application.

46.0.151 How to know how many CPUs are present?

Plugin Version: all, Platform: macOS.

Answer: Try this function:

Example:

```
Function GetCPUCount() as Integer
Declare Function MPProcessors Lib "Carbon" () as Integer
```

```
Return MPProcessors()
End Function
```

Notes: Your app will then need that library to launch on Classic. To avoid this the MBS plugin checks if this library is available and return 1 if it's not available.

46.0.152 How to know the calling function?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac you can use a helper function like this this code:

Example:

```
Public Function CallingFunction() as string
// Query name of calling function of a function

#pragma BreakOnExceptions false

try

// raise a dummy exception
Var r as new NilObjectException
raise r

catch x as NilObjectException

// get stack
Var stack() as string = x.Stack
```

```
// pick function name and return
Var name as string = stack(2)
Return name
```

```
end try
End Function
```

Notes: You need to include function names in your application.

46.0.153 How to launch an app using it's creator code?

Plugin Version: all, Platform: macOS.

Answer: Send an AppleEvent "oapp" with the creator code to the Finder ("MACS"):

Example:

```
Var a as AppleEvent
Var creator as string

creator = "MSIE" ' here the Internet Explorer

a = NewAppleEvent("aevt", "odoc", "MACS")
a.Timeout = -1

a.ObjectSpecifierParam("—") = GetUniqueIDObjectDescriptor("appf", nil, creator)

if not a.send then
msgBox "An error has occured"
else
end if
```

46.0.154 How to launch disc utility?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use this code:

Example:

```
Var f as FolderItem = LaunchServicesFindApplicationForInfoMBS("", "com.apple.DiskUtility", "")

if f<>Nil then
f.Launch
end if
```


Notes: This works even if people renamed the disc utility or moved it to another folder.

46.0.155 How to make a lot of changes to a REAL SQL Database faster?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You may try to embed your changes to the database between two transaction calls.

Example:

```
Var db as Database // some database

db.SQLExecute "BEGIN TRANSACTION"
// Do some Stuff
db.SQLExecute "END TRANSACTION"
```

Notes: This can increase speed by some factors.

46.0.156 How to make a NSImage object for my retina enabled app?

Plugin Version: all, Platform: macOS.

Answer: You can use code like this:

Example:

```
Function NewRetinaImage(pic as Picture, mask as Picture = nil) As NSImageMBS
// first make a NSImageMBS from it
Var n as new NSImageMBS(pic, mask)

// now set to half the size, so we have 2x pixels for the image
n.size = new NSSizeMBS(n.width/2, n.height/2)

// and return
Return n
End Function
```

Notes: The thing to do is to have 2x the pixels, but assign a size to the image which gives it the right size in points.

You can pass the NSImageMBS from here to NSMenuItemMBS. For Retina displays, the full resolution is used. For others it will be reduced.

46.0.157 How to make a window borderless on Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this declares:

Example:

```
// Sets window to borderless popup type, and sets its initial dimensions.
// Call this method, then Win32SetBorderlessPos, and then RB's Show
// method. Use RB Frame type 7 (Global Floating Window).
```

```
Const SWP_NOMOVE = &H2
Const SWP_FRAMECHANGED = &H20
Const HWND_TOPMOST = -1
Const GWL_STYLE = -16
Const WS_POPUPWINDOW = &H80880000
```

```
Var styleFlags as Integer
```

```
#If TargetWin32 Then
```

```
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (hwnd as Integer, nIndex as Integer, dwNewLong as Integer) as Integer
```

```
Declare Function SetWindowPos Lib "user32" (hwnd as Integer, hWndInstertAfter as Integer, x as Integer, y as Integer, cx as Integer, cy as Integer, flags as Integer) as Integer
```

```
styleFlags = SetWindowLong( w.WinHWND, GWL_STYLE, WS_POPUPWINDOW )
styleFlags = BitwiseOr( SWP_FRAMECHANGED, SWP_NOMOVE )
styleFlags = SetWindowPos( w.WinHWND, HWND_TOPMOST, 0, 0, wd, ht, styleFlags )
```

```
#EndIf
```

46.0.158 How to make an alias using AppleEvents?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
Sub MakeAlias(folder as folderitem, target as folderitem, aliasname as string)
```

```
Var ev as AppleEvent
```

```
Var myResult as boolean
```

```
Var properties as AppleEventRecord
```

```
ev = NewAppleEvent("core", "crel", "MACS")
```

```
ev.MacTypeParam("kocl") = "alis"
```

```
ev.FolderItemParam("to ") = target
```

```

ev.FolderItemParam("insh") = folder

properties=new AppleEventRecord
properties.StringParam("pnam")=aliasname

ev.RecordParam("prdt")=properties

myResult = ev.send
// true on success, false on error
End Sub

```

Notes: Call it like this:

```
MakeAlias SpecialFolder.Desktop, SpecialFolder.Desktop.Child("Gif Copy.rb"), "test.rb alias"
```

Seems to not work on Mac OS X 10.6

46.0.159 How to make AppleScripts much faster?

Plugin Version: all, Platform: macOS.

Answer: use "ignoring application responses" like in this example:

```

Notes: on run { fn, fpx, fpy }
ignoring application responses
tell app "Finder" to set the position of folder fn to fpx, fpy
end ignoring
end run

```

46.0.160 How to make double clicks on a canvas?

Plugin Version: all, Platform: macOS.

Answer:

Update: Newer Xojo versions support DoubleClick event, so you don't need this code.

Here's my tip from the tips list on how to add a double-click event to the Canvas control. The technique could easily be used for a window or any Rectcontrol:

Because of its built-in drawing methods, the Canvas control is often used to create custom interface controls. But while the Canvas control has event handlers for most mouse events, it doesn't have an event handler for DoubleClick events. Fortunately, you can add a double-click event handler to a Canvas control easily. Basically, you're going to create a new class based on Canvas and add a double-click event to that. You can

then use the new class anytime you need a Canvas with a double-click event.

To create a new Canvas class with a DoubleClick event handler, do this:

1. Add a new class to your project.
2. Set the Super property of the new class to "Canvas".
3. Change the name of this new class to "DoubleClickCanvas".

A double-click occurs when two clicks occur within the users double-click time (set in the Mouse control panel on both Macintosh and Windows) and within five pixels of each other. So, you'll need a few properties to store when and where the last click occurred.

4. Add a new property with the following declaration and mark it as private: lastClickTicks as Integer
5. Add a new property with the following declaration and mark it as private: lastClickX as Integer
6. Add a new property with the following declaration and mark it as private: lastClickY as Integer

Since the Canvas control doesn't have a DoubleClick event, you will need to add one.

7. Add a new event to your class by choosing New Event from the Edit menu and enter "DoubleClick" as the event name.

Double-clicks occur on MouseUp. In order for the mouseUp event to fire, you must return True in the MouseDown event.

8. In the MouseDown event, add the following code:
Return True

In the MouseUp event, you will need to determine what the users double-click time is. This value is represented on both the Mac and Windows in ticks. A tick is 1/60th of a second. Since there isn't a built-in function for this, you'll need to make a toolbox call. The mouseUp event code below makes the appropriate toolbox call for both Macintosh and Windows. It then compares the time of the users last click to the time of the current click and compares the location of the users last click to the location of the current click.

9. Add the following code to the MouseUp event:

```
Var doubleClickTime, currentClickTicks as Integer

#if targetMacOS then
Declare Function GetDbfTime Lib "Carbon" () as Integer
doubleClickTime = GetDbfTime()
#endif
```

```

#if targetWin32 then
Declare Function GetDoubleClickTime Lib "User32.DLL" () as Integer
doubleClickTime = GetDoubleClickTime()/60 // convert to ticks from milliseconds
#endif

currentClickTicks = ticks
//if the two clicks happened close enough together in time
if (currentClickTicks - lastClickTicks) <= doubleClickTime then
//if the two clicks occurred close enough together in space
if abs(X - lastClickX) <= 5 and abs(Y - LastClickY) <= 5 then
DoubleClick //a double click has occurred so call the event
end if
end if
lastClickTicks = currentClickTicks
lastClickX = X
lastClickY = Y

```

10. Now to test out your new DoubleClickCanvas, drag the class from the Project window to a window in your project to create an instance of it.

11. Double-click on the canvas you just added to your window to open the Code Editor. Notice that the canvas has a DoubleClick event handler. In this event handler, add the following code:

BEEP

46.0.161 How to make my Mac not sleeping?

Plugin Version: all, Platform: macOS.

Answer: Just inform the Mac OS about some system activity with code like this:

Example:

```
Sub UpdateSystemActivity()
```

```

#if TargetCarbon
declare function myUpdateSystemActivity lib "Carbon" alias "UpdateSystemActivity" (activity as Integer)
as short

```

```

const OverallAct = 0 // Delays idle sleep by small amount */
const UsrActivity = 1 // Delays idle sleep and dimming by timeout time */
const NetActivity = 2 // Delays idle sleep and power cycling by small amount */
const HDAActivity = 3 // Delays hard drive spindown and idle sleep by small amount */
const IdleActivity = 4 // Delays idle sleep by timeout time */

```

```
Var e as Integer
```

```
e=myUpdateSystemActivity(UsrActivity)

// you may react on an error if e is not 0 after the call.

#endif
End Sub
```

Notes: You may use another constant if you prefer some different behavior. Call it maybe every second.

46.0.162 How to make my own registration code scheme?

Plugin Version: all, Platform: Windows.

Answer: There are excellent articles about how to make a registration code scheme, but you can also simply use our RegistrationEngineMBS class.

Notes: If you need a license text, why not use the one from Xojo as a starting point?

46.0.163 How to make small controls on Mac OS X?

Plugin Version: all, Platform: macOS.

Answer: You can try this code on Mac OS X:

Example:

```
/*
 * Use the control's default drawing variant. This does not apply to
 * Scroll Bars, for which Normal is Large.
 */
const kControlSizeNormal = 0

/*
 * Use the control's small drawing variant. Currently supported by
 * the Check Box, Combo Box, Radio Button, Scroll Bar, Slider and Tab
 * controls.
 */
const kControlSizeSmall = 1

/*
 * Use the control's small drawing variant. Currently supported by
 * the Indeterminate Progress Bar, Progress Bar and Round Button
 * controls.
 */
```

```

const kControlSizeLarge = 2

'/*
' * Control drawing variant determined by the control's bounds. This
' * ControlSize is only available with Scroll Bars to support their
' * legacy behavior of drawing differently within different bounds.
' */
const kControlSizeAuto = &hFFFF

const kControlSizeTag = "size"

declare function SetControlData lib "Carbon" (controlhandle as Integer, part as short, tagname as OS-
Type, size as Integer, data as ptr) as short

Var m as MemoryBlock

m=NewMemoryBlock(2)
m.UShort(0)=kControlSizeSmall

Title=str(SetControlData(CheckBox1.Handle, 0, kControlSizeTag, 2, m))

```

46.0.164 How to mark my Mac app as background only?

Plugin Version: all, Platform: macOS.

Answer: You can run a build script on each build with this code:

Example:

```

Var App As String = CurrentBuildLocation + "/" + CurrentBuildAppName + ".app"
Call DoShellCommand("/usr/bin/defaults write " + App + "/Contents/Info ""NSUIElement"" YES")

```

Notes: This will set the NSUIElement flag to YES.

46.0.165 How to move a file or folder to trash?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like below:

Example:

```

Function MoveToTrash(f as FolderItem) As Boolean
#If TargetMacOS then
Var r as FolderItem
Var e as Integer = MacFileOperationMBS.MoveObjectToTrashSync(f, r, MacFileOperationMBS.kFSFile-
OperationDefaultOptions)

```

```

if e = 0 then
Return true // Ok
end if

#elseif TargetWin32 then
Var w as new WindowsFileCopyMBS

Var flags as Integer = w.FileOperationAllowUndo + w.FileOperationNoErrorUI + w.FileOperationSilent
+ w.FileOperationNoConfirmation
if w.FileOperationDelete(f, flags) then
Return true // OK
end if

flags = w.FileOperationNoErrorUI + w.FileOperationSilent + w.FileOperationNoConfirmation
if w.FileOperationDelete(f, flags) then
Return true // OK
end if
#else
// Target not supported
break
Return false
#endif
End Function

```

Notes: If you want to move a file to trash, you could use `f.movefileto f.trashfolder`, but that will overwrite existing files in the trash. You can use our `MacFileOperationMBS` class to move a file on Mac to the trash. And it uses the same code as the Finder, so files are renamed when the same name is already in use in the trash:

On Windows we use `WindowsFileCopyMBS` class.
Requires Mac OS X 10.5.

46.0.166 How to move an application to the front using the creator code?

Plugin Version: all, Platform: macOS.

Answer: This makes SimpleText (Code ttxt) to the frontmost application:

Example:

```

Var a as appleevent

a=newappleEvent("misc","actv","ttxt")

if a.send then

```


end if

Notes: (Code is Mac only)

46.0.167 How to move file with ftp and curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can set post/pre quotes to have ftp commands executed before or after the download/upload.

Example:

```
Var d as CURLMBS // your curl object
```

```
// rename/move file
```

```
Var ws() As String
```

```
ws.Append "RNFR Temp.txt"
```

```
ws.append "RNT0 MyFile.txt"
```

```
d.SetOptionPostQuote(ws)
```

Notes: Use SetOptionPostQuote, SetOptionPreQuote or SetOptionQuote.

The ftp commands you pass here are native ftp commands and not the commands you use with ftp applications. So rename is two commands. First RNFR to tell where to rename from and second RNT0 with the new file name. To delete use DELE and the file path.

46.0.168 How to normalize string on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like below:

Example:

```
Function Normalize(t as string) As string
```

```
const kCFStringNormalizationFormD = 0 // Canonical Decomposition
```

```
const kCFStringNormalizationFormKD = 1 // Compatibility Decomposition
```

```
const kCFStringNormalizationFormC = 2 // Canonical Decomposition followed by Canonical Composition
```

```
const kCFStringNormalizationFormKC = 3 // Compatibility Decomposition followed by Canonical Composition
```

```
Var s as CFStringMBS = NewCFStringMBS(t)
```

```
Var m as CFMutableStringMBS = s.Normalize(kCFStringNormalizationFormD)
```

```
Return m.str
```

```
End Function
```

Notes: This uses Apple's CFString functions to normalize unicode variants.

46.0.169 How to obscure the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub ObscureCursor Lib "Carbon" ()
```

```
ObscureCursor
```

Notes: The MBS Plugin has this function, but it's not supported for Windows.

46.0.170 How to open icon file on Mac?

Plugin Version: all, Platform: macOS.

Answer: Use the NSImageMBS class like this:

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.ico")
```

```
Var n as new NSImageMBS(f)
```

```
window1.Backdrop = n.CopyPictureWithMask
```

46.0.171 How to open PDF in acrobat reader?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
Var pdf as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
```

```
// open PDF in Acrobat Reader on Mac:
```

```
// find app
```

```
Var bundleID as string = "com.adobe.Reader"
```

```
Var app as FolderItem = LaunchServicesFindApplicationForInfoMBS("", bundleID, "")
```

```

if app<>nil then

// launch app with parameters

Var docs() as FolderItem
docs.Append pdf

Var param as new LaunchServicesLaunchParameterMBS
param.Defaults = true
param.Application = app

Var x as FolderItem = LaunchServicesOpenXMBS(docs, param)

// on failure, simply launch it
if x = nil then
pdf.Launch(true)
end if

else
pdf.Launch(true)
end if

```

Notes: On Windows, simply use pdf.launch or WindowsShellExecuteMBS.

46.0.172 How to open printer preferences on Mac?

Plugin Version: all, Platform: macOS.

Answer: You can use our OpenMacOSXPreferencesPaneMBS function like this:

Example:

```

Var e as Integer = OpenMacOSXPreferencesPaneMBS("PrintAndFax")
if 0 = e then
MsgBox "OK"
elseif e = -43 then
MsgBox "File not found."
else
MsgBox "Error: "+str(e)
end if

```

46.0.173 How to open special characters panel on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have functions for that in Cocoa and Carbon.

Example:

```
Var a as new NSApplicationMBS
a.orderFrontCharacterPalette
```

Notes: For Cocoa, you can use `orderFrontCharacterPalette` method in `NSApplicationMBS` class.

Or simply for Carbon and Cocoa the `ShowCharacterPaletteMBS` method.

46.0.174 How to optimize picture loading in Web Edition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the `WebPicture` class.

Notes: Take your picture and create a `WebPicture` object. Store this `WebPicture` in a property of the `WebPage`, `Session` or `app` (as global as possible). On the first time you use this picture on an user session, the browser will load it. Second time you use it, the browser will most likely pick it from the cache.

Having pictures in `App` or some module reuses the same picture for all sessions which reduces memory footprint.

This does not work well with pictures you change very often or use only for one webpage on one user.

If you like to see an example, check our `Map` example.

46.0.175 How to parse XML?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
Var s as string = "<test><test /></test>"
```

```
try
Var x as new XmlDocument(s)
MsgBox "OK"
catch xe as XmlException
MsgBox "invalid XML"
end try
```

Notes: If you got an exception, you have a parse error.

46.0.176 How to play audio in a web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the HTML5 audio tag and control it with javascript.

Notes: This is just another example app I made today. It plays a christmas song. The audio file is provided by the application to the server, so no external web server is needed and this application can run stand alone. To compile and run you need Xojo 2010r5.

In the open event we search the audio files and open them as binarystreams. We create the two webfile objects. Those webfiles are part of the app class, so we have them globally. There we set the data with the content of our streams. We also define file names and mime types. They are needed so browser know what we have here:

```
audioFileM4V = new WebFile
audioFileM4V.Data = bM.Read(BM.Length)
audioFileM4V.Filename = "music.m4a"
audioFileM4V.MIMEType = "audio/m4a"
```

```
audioFileOGG = new WebFile
audioFileOGG.Data = bO.Read(BO.Length)
audioFileOGG.Filename = "music.ogg"
audioFileOGG.MIMEType = "audio/ogg"
```

Next in the open event of the webpage we have a PageSource control. The location is set to be before content. In the open event we define the html code for this. First we pick the URLs for the audio files. Than we build the html to use the audio tag. As you see, we give it an ID for later use and have it preload automatically. If you add an autoplay tag, you can have the audio play right away. Inside the audio tag we have two sources so we provide audio for both Firefox (OGG) and Safari (MPEG4). Finally we have a text to display if HTML5 audio tag is not supported.

You can set the source in the EditSource event:

```
Var urlO as string = app.audioFileOGG.URL
Var urlM as string = app.audioFileM4V.URL
me.Source = "<audio id=""mymusic"" preload=""auto""><source src="""+urlO+""" type=""audio/ogg""
/><source src="""+urlM+""" type=""audio/mpeg"" />Your browser does not support the audio ele-
ment.</audio>"
```

Next in the Play button we execute code to play the audio. This is a short javascript code which searches in the html document for the element with the ID "mymusic" which is the ID of our audio tag above. Once we got the object, we call it's play method to start playback.

```
me.ExecuteJavaScript("document.getElementById('mymusic').play();")
```

same for pause:

```
me.ExecuteJavaScript("document.getElementById('mymusic').pause();")
```

and finally for changing volume:

```
me.ExecuteJavaScript("document.getElementById('mymusic').volume="+str(me.Value/100.0)+"");")
```

46.0.177 How to pretty print xml?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the XML Transform method with the right XLS.

Notes: Learn more here:

<http://docs.xojo.com/index.php/XMLDocument.Transform>

46.0.178 How to print to PDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: This code below shows how to redirect printing to a PDF file on Mac OS X.

Example:

```
// get Xojo printer setup
Var p as new PrinterSetup

// now put it into NSPrintInfo to manipulate
Var n as new NSPrintInfoMBS
n.SetupString = p.SetupString

// change destination to file
Var f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
n.SetSaveDestination(f)

// move back
p.SetupString = n.SetupString

// and print as usual
Var g as Graphics = OpenPrinter(p)
g.DrawString "Hello World", 20, 20
```

Notes: And you can use normal graphics class for that.

46.0.179 How to query Spotlight's Last Open Date for a file?

Plugin Version: all, Platform: macOS.

Answer: You can use a MDItemMBS objec to query this value:

Example:

```
Function LastOpenedDate(Extends F As FolderItem, DefaultOtherDates As Boolean = True) As Date
#If TargetMacOS Then
Var xMDItem as New MDItemMBS(F)
Var xDate as Variant

If xMDItem <>Nil Then
xDate = xMDItem.GetAttribute(xMDItem.kMDItemLastUsedDate).DateValue
If xDate IsA Date Then Return xDate
Else
If xDate <>Nil Then Break
End If
#EndIf

If DefaultOtherDates Then
If F.ModificationDate <>Nil Then Return F.ModificationDate
If F.CreationDate <>Nil Then Return F.CreationDate
End If
End Function
```

Notes: Thanks for Josh Hoggan for this example code.

46.0.180 How to quit windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetwin32 then
Var i1,i2,r as Integer
declare function ExitWindowsEx lib "user32" (uFlags as Integer, dwReserved as Integer) as Integer
i1 = 2
i2 = 0
r = ExitWindowsEx(i1,i2)
if r<>0 then
' Error()
```

```
end if
#endif
```

Notes: uFlags parameters:

```
'4 = EWX_Force
'0 = EWX_Logoff
'2 = EWX_Reboot
'1 = EWX_shutdown, should shut down computer
```

Also check the ExitWindowsMBS method.

46.0.181 How to read a CSV file correctly?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With all the rules for quotes and delimiters, you can simply use the SplitCommaSeparatedValuesMBS method in our plugins like this:

Example:

```
Var f as FolderItem = SpecialFolder.Desktop.Child("test.csv")
Var t as TextInputStream = f.OpenAsTextFile

while not t.EOF
Var s as string = t.ReadLine(encodings.ASCII)

Var items() as string = SplitCommaSeparatedValuesMBS(s, ";", """")

List.AddRow ""
Var u as Integer = UBound(items)
for i as Integer = 0 to u
List.Cell(List.LastIndex,i) = items(i)
next

wend
```

Notes: Please make sure you choose the right text encoding.

46.0.182 How to read the command line on windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetwin32 then
Var line as string
Var mem as MemoryBlock

Declare Function GetCommandLineA Lib "kernel32" () As Ptr

mem=GetCommandLineA()
s=mem.cstring(0)

#endif
```

Notes: Newer Xojo versions have a system.commandline property.

46.0.183 How to render PDF pages with PDF Kit?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
// choose a file
Var f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")

// open it as PDF Document
Var sourceFile as New PDFDocumentMBS(f)

if sourceFile.handle <>0 then // it is a PDF file

// get upper bound of pages
Var c as Integer = sourceFile.pageCount-1

// from first to last page
for n as Integer = 0 to c

// pick that page
Var page as PDFPageMBS = sourceFile.pageAtIndex(n)

// render to image
Var p as NSImageMBS = page.Render

// and convert to RB picture and display
Backdrop = p.CopyPictureWithMask
```

next

end if

Notes: PDFKit works only on Mac OS X.

46.0.184 How to restart a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
Var ae as appleevent
ae=newappleEvent("FNDR","rest","MACS")
if not ae.send then
msgBox "The computer couldn't be restarted."
end if
```

46.0.185 How to resume ftp upload with curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: CURL supports that and you simply need to set the right options.

Notes: First of course OptionUpload must be true. Second OptionFTPAppend must be true so the OptionResumeFrom is used. Store there (or in OptionResumeFromLarge) your start value.

Don't forget to implement the read event and return data there as requested.

46.0.186 How to rotate a PDF page with CoreGraphics?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: This code opens a PDF and draws the first page into a new PDF with 90–∞ rotation.

Example:

```
// Rotate a PDF page

// our files
Var sourcefile as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
Var destfile as FolderItem = SpecialFolder.Desktop.Child("rotated.pdf")

// open PDF
Var pdf as CGPDFDocumentMBS = sourcefile.OpenAsCGPDFDocumentMBS
```

```

// query media size of first page
Var r as CGRectMBS = pdf.MediaBox(1)

// create new PDF
Var c as CGContextMBS = destfile.NewCGPDFDocumentMBS(r,"title","Author","Creator")

// create rotated rectangle
Var nr as new CGRectMBS(0,0,r.Height,r.Width)

// create new page
c.BeginPage nr
c.SaveGState

const pi = 3.14159265

// rotate by 90-∞
c.RotateCTM pi*1.5

// fix origin
c.TranslateCTM -r.width,0

// draw PDF
c.DrawCGPDFDocument pdf,r,1

// cleanup
c.RestoreGState
c.EndPage

c = nil

// show in PDF viewer
destfile.Launch

```

Notes: This code is Mac only as it needs CoreGraphics.

46.0.187 How to rotate image with CoreImage?

Plugin Version: all, Platform: macOS.

Answer: Use the code like the one below:

Example:

```

// Rotate image with CoreImage

// load image

```

```

Var f as FolderItem = SpecialFolder.Desktop.Child("test.png")
Var image as new CIIImageMBS(f)

// rotate 45 degree
Var n as new NSAffineTransformMBS
n.rotateByDegrees(45)

Var TransformFilter as new CIFilterAffineTransformMBS
TransformFilter.inputImage = image
TransformFilter.inputTransform = n

// get result
Var resultImage as CIIImageMBS = TransformFilter.outputImage

// for saving to file
Var outputImage as NSImageMBS = resultImage.RenderNSImage(false)

f = SpecialFolder.Desktop.Child("output.png")
Var b as BinaryStream = BinaryStream.Create(f, true)
b.Write outputImage.PNGRepresentation

// as Xojo picture object for display
Var pic as Picture = outputImage.CopyPictureWithMask

Backdrop = pic

```

46.0.188 How to run a 32 bit application on a 64 bit Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Install 32 bit compatibility libraries.

Notes: The package is called ia32-libs for ubuntu (and others).

Some applications need to be run on a 32 bit system as they need some hardware related libraries. Like libUSB or libHID for USB devices.

46.0.189 How to save HTMLViewer to PDF with landscape orientation?

Plugin Version: all, Platform: macOS.

Answer: You can use NSPrintInfoMBS to change the options for PrintToPDFFile function.

Example:

```

// make it landscape
Var n as NSPrintInfoMBS = NSPrintInfoMBS.sharedPrintInfo
n.orientation = n.NSLandscapeOrientation

```

```
// save html to file
Var f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
call HTMLViewer1.PrintToPDFFileMBS(f,10,30,10,30)
```

Notes: You may want to reset options later.
This code is only for Mac OS X.

46.0.190 How to save RTFD?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: How to load PDF to htmlviewer on desktop?

Example:

```
Public Sub LoadPDFData(viewer as HTMLViewer, PDFData as string)
Var base64string As String = EncodeBase64(PDFData)

// remove line endings to make it a big line
base64string = ReplaceLineEndings(base64string, "")

// build data URL
// https://en.wikipedia.org/wiki/Data_URI_scheme
Var dataURL As String = "data:application/pdf;base64," + base64string

// show in webviewer
HTMLViewer1.LoadURL(dataURL)

// may not work everywhere due to URL length limit
// for Web projects, use WebFile instead!
End Sub
```

Notes: This avoids a temporary file, which may also work.
For Web Apps, please use WebFile.

46.0.191 How to save RTFD?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With NSTextViewMBS you can use this code to save to RTFD:

Example:

```
// save text as RTFD including image attachments
Var f as FolderItem = GetSaveFolderItem(FileTypes1.ApplicationRtf, "test.rtf")
```

```

if f = nil then Return

Var a as NSAttributedStringMBS = textView.textStorage
Var w as NSFileWrapperMBS = a.RTFDFileWrapperFromRange(0, a.length, DocumentAttributes)

Var e as NSErrorMBS
if w.writeToFile(f, e) then

else
MsgBox e.LocalizedDescription
end if

```

Notes: For TextArea you can query the underlying NSTextViewMBS object via TextArea.NSTextViewMBS method.

46.0.192 How to scale a picture proportionally with mask?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: For a proportional scaling, we calculate the new picture size relative to the target maximum size.

Example:

```

Function ProportionalScaledWithMask(extends pic as Picture, Width as Integer, Height as Integer) As Picture
// Calculate scale factor

Var faktor as Double = min( Height / Pic.Height, Width / Pic.Width)

// Calculate new size
Var w as Integer = Pic.Width * faktor
Var h as Integer = Pic.Height * faktor

// create new picture
Var NewPic as new Picture(w,h,32)

// check if we have a mask and clear it
Var m as picture = pic.mask(False)
pic.mask = nil

// draw picture in the new size
NewPic.Graphics.DrawPicture Pic, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height

if m <>nil then
// restore mask and scale it
pic.mask = m

```

```
NewPic.mask.Graphics.DrawPicture m, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height
end if
```

```
// return result
Return NewPic
End Function
```

Notes: This version handles mask. As you see we actually have to remove mask in order to copy the picture part correctly.

46.0.193 How to scale a picture proportionally?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: For a proportional scaling, we calculate the new picture size relative to the target maximum size.

Example:

```
Function ProportionalScaled(extends pic as Picture, Width as Integer, Height as Integer) As Picture
// Calculate scale factor

Var faktor as Double = min( Height / Pic.Height, Width / Pic.Width)

// Calculate new size
Var w as Integer = Pic.Width * faktor
Var h as Integer = Pic.Height * faktor

// create new picture
Var NewPic as new Picture(w,h,32)

// draw picture in the new size
NewPic.Graphics.DrawPicture Pic, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height

// return result
Return NewPic
End Function
```

Notes: This does not handle mask, but you can scale the mask the same way and assign it to the new picture.
(see other FAQ entry with mask)

46.0.194 How to scale/resize a CImageMBS?

Plugin Version: all, Platform: Windows.

Answer: Use the CIFilterLanczosScaleTransform filter to scale down a picture to a specific size.

Example:

```

Var pic As Picture = LogoMBS(500)
Var image As CImageMBS = CImageMBS.imageWithPicture(pic)

Var filter As New CIFilterLanczosScaleTransformMBS

Const targetWidth = 600.0
Const targetHeight = 400.0

Var scale As Double = targetHeight / image.Extent.Height
Var aspect As Double = targetWidth / (image.Extent.Width * scale)

filter.inputImage = image
filter.inputScale = scale
filter.inputAspectRatio = aspect

Var result As Picture = filter.outputImage.RenderPicture

Backdrop = result

```

Notes: This is same code as our scaleTo convenience method.

46.0.195 How to scale/resize a picture?

Plugin Version: all, Platform: Windows.

Answer: There are several ways to scale or resize a picture. The easiest way may be the ScaleMBS function in the Picture class.

Example:

```

Var Original,Scaled as Picture

Original=LogoMBS(500)
Scaled=Original.ScaleMBS(100,100,true)

```

Notes: The plugin ways:

- GraphicsMagick can scale/resize.
- CoreImage scale filter may result in the fastest and best images on Mac OS X 10.4.
- NSImageMBS can scale, but is Mac OS X only.
- CGImageMBS can scale, but is Mac OS X only.

- CImageMBS can scale, but is Mac OS X only.
- QuickTime Graphics exporter and importer can be connected to scale. (this was used more often a few years ago)
- ImageMagick can scale very nice and crossplatform. But the ImageMagick libraries are big.
- The picture.ScaleMBS function is self written and results in equal output on Mac, Windows and Linux without any additional libraries installed.
- Picture.ScalingMBS does crossplatform scaling with several modes.

with pure Xojo:

- make a new picture and draw the old one with new size inside.

46.0.196 How to search with regex and use unicode codepoints?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can specify unicode characters in search string with backslash x and digits.

Example:

```
Var r as RegExMbs
Var s as string
Var c as Integer
```

```
s="123 √§√√° ABC 456"
```

```
r=new RegExMBS
if r.Compile("√.") then
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if
```

```
r=new RegExMBS
if r.Compile("\xF6.") then // finds √ using Unicode codepoint
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if
```

46.0.197 How to see if a file is invisible for Mac OS X?

Plugin Version: all, Platform: macOS.

Answer: Try this function:

Example:

```
Function Invisible(F As FolderItem) As Boolean
Var TIS As TextInputStream
Var S,All As String
Var I as Integer
Var g as folderitem
```

```
If Left(F.Name,1)="." or not f.visible Then
Return True
End If
```

```
g=F.Parent.Child(".hidden")
If g.Exists Then
TIS=g.OpenAsTextFile
if tis<>Nil then
All=TIS.ReadAll
For I=1 to CountFields(All,Chr(11))
S=NthField(All, Chr(11), I)
If S=F.name Then
Return True
End If
Next
end if
End if
End Function
```

46.0.198 How to set cache size for SQLite or REALSQLDatabase?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You use the pragma cache_size command on the database.

Example:

```
// set cache size to 20000 pages which is about 20 MB for default page size
Var db as REALSQLDatabase
db.SQLExecute "PRAGMA cache_size = 20000"
```

Notes: Default cache size is 2000 pages which is not much.

You get best performance if whole database fits in memory.

At least you should try to have a cache big enough so you can do queries in memory.

You only need to call this pragma command once after you opened the database.

46.0.199 How to set the modified dot in the window?

Plugin Version: all, Platform: macOS.

Answer: Try this declares:

Example:

```
window1.ModifiedMBS=true
```

46.0.200 How to show a PDF file to the user in a Web Application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use a WebHTMLViewer control and load the

Example:

```
Var CurrentFile as WebFile // a property of the WebPage

// define the PDF file
CurrentFile = new WebFile
CurrentFile.Filename = "test.pdf"
CurrentFile.MIMEType = "application/pdf"
CurrentFile.Data = "some pdf data" // MyDynaPDF.GetBuffer

// load into html viewer
HTMLViewer1.URL = CurrentFile.URL
```

Notes: See our Create PDF example for the Xojo Web Edition.

46.0.201 How to show Keyboard Viewer programmatically?

Platform: macOS.

Answer: Use Xojo or AppleScript to launch the KeyboardViewerServer.app.

Example:

```
Var a as new AppleScriptMBS
Var text as string
Var lines(-1) as string

lines.append "set theApplication to ""KeyboardViewerServer""
lines.append "set thePath to ""/System/Library/Components/KeyboardViewer.component/Contents/Shared-
Support/KeyboardViewerServer.app"""
```

```

lines.append ""
lines.append "set POSIXPath to ((POSIX file thePath) as string)"
lines.append "tell application ""System Events"" to set isRunning to 0 <(count (application processes whose
name is theApplication))"
lines.append "if isRunning then tell application POSIXPath to quit"
lines.append "delay 0.15"
lines.append ""
lines.append "ignoring application responses"
lines.append " tell application POSIXPath to run"
lines.append "end ignoring"

```

```
text=join(lines,EndOfLine.macintosh)
```

```

a.Compile text
a.Execute

```

Notes: AppleScript code:

```

set theApplication to "KeyboardViewerServer"
set thePath to "/System/Library/Components/KeyboardViewer.component/Contents/SharedSupport/Key-
boardViewerServer.app"

```

```

set POSIXPath to ((POSIX file thePath) as string)
tell application "System Events" to set isRunning to 0 <(count (application processes whose name is theAp-
plication))
if isRunning then tell application POSIXPath to quit
delay 0.15

```

```

ignoring application responses
tell application POSIXPath to run
end ignoring

```

46.0.202 How to show the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub ShowCursor Lib "Carbon" ()
```

```
ShowCursor
```

Notes: The MBS Plugin has this function and supports it on Windows, too.

46.0.203 How to shutdown a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
Var ae as appleevent
ae=newappleEvent("FNDR","shut","MACS")
if not ae.send then
msgBox "The computer couldn't be shutdown."
end if
```

Notes: Or toolbox call (Attention: This method will stop the computer immediatly: No document asked to be saved, all applications quitting without knowing).

```
Declare Sub ShutDwnPower Lib "Carbon" ()
ShutDwnPower
```

46.0.204 How to sleep a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
Var ae as appleevent
ae=newappleEvent("FNDR","slep","MACS")
if not ae.send then
msgBox "The computer doesn't want to sleep."
end if
```

46.0.205 How to speed up rasterizer for displaying PDFs with DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Here a few speed tips:

Notes:

- Use the DynaPDFRasterizerMBS function instead of our render functions.

- Reuse DynaPDFRasterizerMBS as long as the target picture size doesn't change.
- Import only the PDF pages you want to display.
- Let DynaPDF do zooming, rotating or other effects instead of you change it.

46.0.206 How to use PDFLib in my RB application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The PDFlib plugin was discontinued in favor of our DynaPDF plugin.

Notes: If you need help to move, please contact us.

46.0.207 How to use quotes in a string?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Just double them.

Example:

```
msgbox "This String contains ""quotes"""
```

46.0.208 How to use Sybase in Web App?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use our MBS Xojo SQL Plugin to connect to a Sybase Database in your web application.

Notes: If you see db.Connect giving the error message "cs_ctx_alloc ->CS_MEM_ERROR", than some things are not setup right for Sybase.

The Apache process may not have all the SYBASE environment variables being set when the CGI was launched.

Adding these lines to /etc/httpd/conf/httpd.conf stopped the faux memory errors for us:

```
SetEnv LD_LIBRARY_PATH /opt/sybase/OCS-15_0/lib:/opt/sybase/OCS-15_0/lib3p64:/opt/sybase/OCS-15_0/lib3p:  
SetEnv SYBROOT /opt/sybase  
SetEnv SYBASE_OCS /opt/sybase  
SetEnv SYBASE /opt/sybase
```

46.0.209 How to use the Application Support folder?

Plugin Version: all, Platform: macOS.

Answer:

I was saving a registration code for an app to the Preference folder. People on the list have suggested that it would be better in the ApplicationSupportFolder. How do I save the file called CWWPrefs into that folder using MBS?

I have checked for examples and the docs but can't see how to apply it

```
//f = SpecialFolder.Preferences.child("CWWPrefs")
f = ApplicationSupportFolderMBS(-32768)
```

Example:

```
Var folder,file as FolderItem
```

```
folder = createApplicationSupportFolderMBS(-32763)
```

```
if folder=nil then
// Some very old Mac OS Versions may not support it
// or the plugin may fail for any reason
folder=SpecialFolder.Preferences
end if
```

```
file=folder.Child("CWWPrefs")
```

```
MsgBox file.NativePath
```

Notes:

You may not be able to write there with a normal user account!

46.0.210 How to use the IOPMCopyScheduledPowerEvents function in Xojo?

Plugin Version: all, Platform: macOS.

Answer: You can use the following code which does this using the SoftDeclareMBS class.

Example:

```
Sub Open()
Var c as CFDateMBS
Var t as CFAbsoluteTimeMBS
```

```

// get current date
c=NewCFDateMBS

// in absolute time (seconds since x)
t=c.AbsoluteTime

// add 600 seconds (= 10 Minutes)
t.Value=t.Value+600

// Make a Date from it
c=t.Date

// Schedule the event
// 0 on success
// E00002C1 for missing root rights
Title=hex(schedulePowerEvent(c, "wake"))

// Just for information, display the scheduled stuff
CFShowMBS CopyScheduledPowerEvents
End Sub

Function CopyScheduledPowerEvents() As cfarrayMBS
Var s as SoftDeclareMBS
Var m as MemoryBlock

s=new SoftDeclareMBS

if s.LoadLibrary("IOKit.framework") then
if s.LoadFunction("IOPMCopyScheduledPowerEvents") then
if s.CallFunction(0,nil) then
Return NewCFArrayMBSHandle(s.Result,true)
else
MsgBox "Failed to Call IOPMCopyScheduledPowerEvents."
end if
else
MsgBox "Failed to load IOPMCopyScheduledPowerEvents."
end if
else
MsgBox "Failed to load IOKit."
end if

Return nil
End Function

Function SchedulePowerEvent(time_to_wake as CFDateMBS, Type as CFStringMBS) as Integer
Var s as SoftDeclareMBS
Var m as MemoryBlock

```



```

'/*
' * Types of power event
' * These are potential arguments to IOPMSchedulePowerEvent().
' * These are all potential values of the kIOPMPowerEventTypeKey in the CFDictionaryes
' * returned by IOPMCopyScheduledPowerEvents().
' */
'/*!
'@define kIOPMAutoWake
'@abstract Value for scheduled wake from sleep.
' */
'#define kIOPMAutoWake "wake"
,

'/*!
'@define kIOPMAutoPowerOn
'@abstract Value for scheduled power on from off state.
' */
'#define kIOPMAutoPowerOn "poweron"
,

'/*!
'@define kIOPMAutoWakeOrPowerOn
'@abstract Value for scheduled wake from sleep, or power on. The system will either wake OR
'power on, whichever is necessary.
' */
,

'#define kIOPMAutoWakeOrPowerOn "wakepoweron"
'/*!
'@define kIOPMAutoSleep
'@abstract Value for scheduled sleep.
' */
,

'#define kIOPMAutoSleep "sleep"
'/*!
'@define kIOPMAutoShutdown
'@abstract Value for scheduled shutdown.
' */
,

'#define kIOPMAutoShutdown "shutdown"

s=new SoftDeclareMBS

if s.LoadLibrary("IOKit.framework") then
if s.LoadFunction("IOPMSchedulePowerEvent") then

m=NewMemoryBlock(12)
m.Long(0)=time_to_wake.handle
m.Long(4)=0 // nil
m.Long(8)=type.Handle

```

```

if s.CallFunction(3,m) then
Return s.Result
end if
end if
end if

```

End Function

Notes: Requires Mac OS X and to execute root rights.

46.0.211 How to validate a GUID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use this function below which uses a regular expression to verify that the string is a valid UUID/GUID:

Example:

Function IsGUID(guid as string) As Boolean

Var r as new RegEx

```

r.SearchPattern = "^(\{ { 0,1 } ( [ 0-9a-fA-F ] ) { 8 } -( [ 0-9a-fA-F ] ) { 4 } -( [ 0-9a-fA-F ] ) { 4 }
-( [ 0-9a-fA-F ] ) { 4 } -( [ 0-9a-fA-F ] ) { 12 } \} { 0,1 } )$ "

```

Return r.Search(guid)<>nil

End Function

Notes: Simply parsing the GUID with CFUUIDMBS does not give the same result as CFUUIDMBS will also take a string like "DDDD".

46.0.212 How to walk a folder hierarchie non recursively?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this one:

Example:

Sub Walk(folder as FolderItem)

Var folders() as FolderItem

folders.Append folder

while UBound(folders)>=0

```

Var currentFolder as FolderItem = folders.pop

Var c as Integer = currentFolder.Count
for i as Integer = 1 to c
Var item as FolderItem = currentFolder.TrueItem(i)

if item = Nil then
// no permission
elseif item.Visible then // only visible

if item.Directory then
folders.Append item
else
// work with file here
end if

end if

next

wend
End Sub

```

Notes: As you see we go with a long loop which runs until we don't have more folders to process.

We ignore items we can't access due to permission limits.

And we only work visible items.

If you like, check `folderitem.isBundleMBS` on item to handle packages and applications better on Mac OS X.

46.0.213 I got this error: PropVal, QDPictMBS.Name (property value), Type mismatch error. Expected CGDataProviderMBS, but got Variant, Name:QDPictMBS

Plugin Version: all, Platform: macOS.

Answer: The plugins MacOSX and MacOSXCF belong together. If you use one part, please also install the other part.

Notes: We splitted the plugin because the Xojo IDE on Windows crashed on compilation.

46.0.214 I registered the MBS Plugins in my application, but later the registration dialog is shown.

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: There are two main reasons.

Notes: 1. you may use the plugin before registering them. This is often the case if you register in a window open event and use the plugin in a control open event.

On the console on Mac OS X or Windows, you may see a message like this "MBS Plugins were used by the application before the RegisterMBSPlugin function was called. Please fix this in your code!".

2. you may have mixed different plugin versions which are not compatible.

In this case you can see a message "Internal plugin registration error." on the console on Mac OS X. Newer plugins may show a message dialog reporting this. Older version simply think they are not registered.

If the installer just merges old and new applications, users may have libraries of older and newer plugin versions in the libs folder. If your application loads the wrong version, the registration fails.

If you use remote debugging, make sure you clear the temporary files there, too. Otherwise you may have old DLLs on your hard disc which may disturb your application.

You can run into issues if you use your registration code on different places of your app. Please register only once in app.open (or app Constructor). If you have several codes, simply call them one after the other.

Also check that you only call RegisterMBSPlugin with valid serial number. If you later call RegisterMBSPlugin with Demo like in example code above, you remove the license.

Next check if you can clear the Xojo caches and that helps. This includes the Xojo Scratch folder and the Plugins & Project caches. Simply locate those folders and delete them. For Windows look in hidden AppData folder in your user folder. For Mac, please check textasciitilde /Library/Caches and your temp folders.

Finally make sure you use the right serial number. Not an older one or a misspelled one.

46.0.215 I want to accept Drag & Drop from iTunes

Plugin Version: all, Platform: macOS.

Answer: You need to accept AcceptMacDataDrop "itun" and Handle the DropObject.

Example:

```
Sub Open()
window1.AcceptMacDataDrop "itun"
End Sub
```

```
Sub DropObject(obj As DragItem)
Var s as string
```

```

Var f as folderItem
Var d as CFDictionaryMBS
Var o as CObjectMBS
Var key as CFStringMBS
Var dl as CFDictionaryListMBS
Var i,c as Integer
Var u as CFURLMBS
Var file as FolderItem

if obj.MacDataAvailable("itun") then
s = obj.MacData("itun")

// Parse XML
o=NewCObjectMBSFromXML(NewCFBinaryDataMBSStr(s))

// Make dictionary
if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)

// get Tracks Dictionary
key=NewCFStringMBS("Tracks")
o=d.Value(key)

if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)
dl=d.List

// Walk over all entries in the Tracks dictionary
c=dl.Count-1
for i=0 to c
o=dl.Value(i)

if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)

key=NewCFStringMBS("Location")
o=d.Value(key)
if o isa CFStringMBS then
u=NewCFURLMBS CFStringMBS(CFStringMBS(o),nil)

file=u.file
if file<>nil then
MsgBox file.NativePath
end if
end if
end if
next
end if

```

```

end if
end if
End Sub

```

Notes: The code above inside a window on Xojo 5.5 with MBS Plugin 5.3 will do it nice and show the paths.

46.0.216 I'm drawing into a listbox but don't see something.

Plugin Version: all.

Answer: If you draw this in a listbox cellbackground, you need to draw on the correct position

Example:

```

Function CellBackgroundPaint(g As Graphics, row as Integer, column as Integer) As Boolean
Var f as FolderItem
f=SpecialFolder.Desktop
f.DrawWideIconMBS(g,listbox1.left,listbox1.top+row*20,16)
Return true
End Function

```

Notes: Try this in a listbox. The Graphics object there has a clipping and an offset which the plugin doesn't know about.

46.0.217 I'm searching for a method or so to move a window from position x.y to somewhere else on the screen.

Platform: macOS.

Answer:

The code I produced in RB isn't smooth enough. Is there a call in MBS, if not, can it be done? The speed of it has to be like the show of a DrawerWindow.

Try the declare below for Carbon. With WindowLib it will work on Mac OS 8.5 and newer.

Notes:

See Window.Transition functions.

46.0.218 If I use one of your plug-ins under windows, would this then impose the use of dll after compilation or my would my compiled soft still be a stand-alone single file software?

Platforms: macOS, Linux, Windows.

Answer: Stand alone.

Notes: Xojo compiles all used plugins into the application binary.

Some plugin parts need external dlls but you will find that in the documentation. (e.g. pdflib for some classes)

46.0.219 Is the fn key on a powerbook keyboard down?

Plugin Version: all, Platform: macOS.

Answer: I am unable to figure out how or if it is possible to detect if the fn key is down on a powerbook keyboard. Is it possible?

Example:

' [Window.Open Event of a blank project](#):

Var i as Integer

```
for i=0 to 127
if keyboard.asynckeydown(i) then
title=str(i) // found
return
end if
next
title="" // not found
```

Notes: This test application shows the keycode (decimal) 63 for the fn key.

46.0.220 Is there a case sensitive Dictionary?

Plugin Version: all.

Answer: The MBS Plugin has several classes which can work as a replacement.

Notes: First you could use VariantToVariantHashMapMBS or VariantToVariantOrderedMapMBS.

If you know that all keys are Strings or Integers only, you can use the specialized classes which are a little bit faster due to avoiding variants:

IntegerToIntegerHashMapMBS class

IntegerToIntegerOrderedMapMBS class

IntegerToStringHashMapMBS class

IntegerToStringOrderedMapMBS class
 IntegerToVariantHashMapMBS class
 IntegerToVariantOrderedMapMBS class
 StringToStringHashMapMBS class
 StringToStringOrderedMapMBS class
 StringToVariantHashMapMBS class
 StringToVariantOrderedMapMBS class

46.0.221 Is there a way to use the MBS plugin to get only the visible item and folder count on a volume?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the DirectorySizeMBS class for this as in the example below:

Example:

```
Var d as DirectorySizeMBS
```

```
d=new DirectorySizeMBS
```

```
// volume(1) as my boot volume is very full
if d.update(volume(1),true,0) then
MsgBox str(d.VisibleItemCount)+" visible items, "+str(d.HiddenItemCount)+" invisible items."
end if
```

Notes: Complete Question: Is there a way to use the MBS plugin to get only the visible item and folder count on a volume? The FileCount and FolderCount properties of VolumeInformationMBS seem to provide the total # of items including invisible items such as .DS_Store and more importantly .Trashes which is causing me a great amount of difficulty during a recursive scan of a volume. I've got a progress bar which uses the total of the filecount and foldercount properties as the maximum value, but my routine needs to filter out all invisible items, as it is creating a catalog of a volume for archiving purposes. Any thoughts how I could get accurate number.

46.0.222 Is there an easy way I can launch the Displays preferences panel?

Plugin Version: all, Platform: macOS.

Answer: Use the code below:

Example:

```
Var error as Integer
```



```

error=OpenMacOSXPreferencesPaneMBS("Displays")
if error<>0 then
MsgBox "Failed to launch QuickTime System Preferences panel."
end if

```

46.0.223 List of Windows Error codes?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have a list of windows error codes on our website.

Notes: <http://www.monkeybreadsoftware.de/xojo/winerror.shtml>

46.0.224 Midi latency on Windows problem?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The issue is system related, not a problem with RB or the plugin.

Notes: Two things will adversely affect the timing:

(1) latency of the software synthesizer output driver. The default Windows wavetable synthesizer has considerable latency. I don't know how many milliseconds, but it is noticeable.

(2) latency of the digital audio output driver. Different systems have different drivers for different audio hardware. My Dell laptop has a minimum 15ms latency in the audio driver.

These two things put together were causing a very sluggish MIDI response. I was able to verify these as the culprits by routing MIDI directly out of RB into a sample player, which only introduces the latency of (2) and does not include latency of (1).

I don't know how widely known are these facts, if not then you may want to add this information to the documentation, since Windows programmers using the MIDI plugin may not know those problems, and might mistakenly blame your plugin, as I did :) Sorry about that!

(From Aaron Andrew Hunt)

46.0.225 My Xojo Web App does not launch. Why?

Plugin Version: all, Platform: macOS.

Answer: Here is a list of checks to do for linux apache installations with Xojo or Xojo Web applications:

Notes: Just a list of checks to do for linux apache installations:

- You have 64bit linux? Than you need 32 bit compatibility libraries.
- The folder of your app is writable? Set permissions to 777.
- The cgi script is executable? Set permissions to 755.
- The app file itself is executable? Set permissions to 755.
- You uploaded cgi file as text, so it has unix line endings? (this often gives error "Premature end of script headers" in apache log)
- You uploaded config.cfg file and made it writable? Set permissions to 666.
- Your apache allows execution of cgi scripts? You enabled cgi for apache and uncommented addhandler command for CGI on a new apache installation?
- You uploaded the app file and libraries as binary files? Upload as text breaks them.
- You did upload the libs folder?
- You don't have code in app.open, session.open and other events which crashes app right at launch?
- You don't have a print command in your app.open event? (see feedback case 23817)
- You allowed htaccess file to overwrite permissions?

46.0.226 SQLiteDatabase not initialized error?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Before you can use SQLiteDatabaseMBS, it must be initialized.

Example:

```
Var d as new SQLiteDatabaseMBS
```

Notes: This happens normally when you use "new SQLiteDatabaseMBS".

But if you just have a SQLConnectionMBS and get a recordset there, the initialization may not have happened, yet.

So please simply add a line "dim d as new SQLiteDatabaseMBS" to your app.open code after registration, so the plugin part can initialize and late provide recordsets.

46.0.227 Textconverter returns only the first x characters. Why?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

Some older Xojo versions limit the Textconverter to around 1024 characters in input and output. This should be fixed with RB5.

Notes:

Xojo seems not to support Textconverters at all on Windows.

46.0.228 The type translation between CoreFoundation/Foundation and Xojo data types.

Plugin Version: all, Platform: macOS.

Answer: The plugin does conversion between Cocoa/Carbon data types and native Xojo data types. The following list help you knowing what the current plugins support:

Notes: Cocoa NSObject to Variant:

```

nil ->nil
NSDictionary ->Dictionary
NSData ->MemoryBlock
NSString ->String
NSAttributedString ->NSAttributedStringMBS
NSDate ->Date
NSNumber ->double/integer/Int64/UInt64/UInt32/Boolean
NSURL ->String
NSValue with NSRect ->NSRectMBS
NSValue with NSPoint ->NSPointMBS
NSValue with NSSize ->NSSizeMBS
NSValue with NSRange ->NSRangeMBS
NSValue with QTTime ->QTTimeMBS
NSValue with QTTimeRange ->QTTimeRangeMBS
NSArray ->Array of Variant
QuartzFilter ->QuartzFilterMBS

```

- ->*MBS

Variant to Cocoa NSObject:

```

nil ->nil
Dictionary ->NSDictionary
Boolean ->NSNumber
Integer ->NSNumber
Color ->NSColor
Int64 ->NSNumber
Single ->NSNumber
Double ->NSNumber

```

Date ->NSDate
MemoryBlock ->NSData
String ->NSString
NSImageMBS ->NSImage
NSAttributedStringMBS ->NSAttributedString
NSColorMBS ->NSColor
NSRectMBS ->NSValue with NSRect
NSSizeMBS ->NSValue with NSSize
NSPointMBS ->NSValue with NSPoint
NSRangeMBS ->NSValue with NSRange
NSBurnMBS ->NSBurn
NSViewMBS ->NSView
NSFontMBS ->NSFont
NSParagraphStyleMBS ->NSParagraphStyle
NSAttributedStringMBS ->NSAttributedString
WebPolicyDelegateMBS ->WebPolicyDelegate
WebUIDelegateMBS ->WebUIDelegate
WebFrameLoadDelegateMBS ->WebFrameLoadDelegate
WebResourceLoadDelegateMBS ->WebResourceLoadDelegate
NSIndexSetMBS ->NSIndexSet
QTTimeMBS ->QTTime
QTTimeRangeMBS ->QTTimeRange
Array of Variant ->NSArray
Array of String ->NSArray
CFStringMBS ->NSString
CFNumberMBS ->NSNumber
CFDataMBS ->NSData
CFURLMBS ->NSURL
CFArrayMBS ->NSArray
CFDictionaryMBS ->NSDictionary
CFBinaryDataMBS ->NSDate

Carbon CTypeRef to Variant:

CFDictionaryRef ->Dictionary
CFStringRef ->String
CFDataRef ->String
CFURL ->String
CFNumber ->Integer/Double/Int64
CFArray ->Array
CFDate ->date
nil ->nil
CGColorSpace ->CGColorSpaceMBS
CGColor ->CGColorMBS
CGImage ->CGImageMBS
CF* ->CF*MBS

Variant to Carbon CFTypeRef:

Dictionary ->CFDictionaryRef
 Boolean ->CFBooleanRef
 Color ->CFNumberRef
 Integer ->CFNumberRef
 Int64 ->CFNumberRef
 Single ->CFNumberRef
 Double ->CFNumberRef
 String ->CFStringRef
 Color ->CGColorRef
 Date ->CFDateRef
 nil ->nil
 Memoryblock ->CFDataRef
 FolderItem ->CFURLRef
 Dictionary ->CFDictionaryRef
 Array of Variant/String/Date/Double/Single/Int64/Integer ->CFArray
 CGRectMBS ->CGRect as CFDataRef
 CGSizeMBS ->CGSize as CFDataRef
 CGPointMBS ->CGPoint as CFDataRef
 CGColorMBS ->CGColor
 CGColorSpaceMBS ->CGColorSpace
 CGImageMBS ->CGImage
 CGDataConsumerMBS ->CGDataConsumer
 CGDataProviderMBS ->CGDataProvider
 CF*MBS ->CF*

Strings without encodings should be put into dictionaries as memoryblocks.

46.0.229 Uploaded my web app with FTP, but it does not run on the server!

Plugin Version: all, Platform: Windows.

Answer: If you see errors like a simple "Segmentation Fault" on Linux or some other wired errors, you may want to check your FTP upload mode. It must be binary for web apps. ASCII mode corrupts the application.

46.0.230 What classes to use for hotkeys?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use CarbonHotKeyMBS class on Mac and WindowsKeyFilterMBS on Windows.

Notes: CarbonHotKeyMBS will also work fine in Cocoa apps.

46.0.231 What do I need for Linux to get picture functions working?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In order to get our plugins working on Linux systems without GUI, the plugin loads graphics libraries dynamically.

Notes: To get it working, the plugin tries to load gtk with this paths:

- libgtk-x11-2.0.so”
- libgtk-x11-2.0.so.0”
- /usr/lib/libgtk-x11-2.0.so”
- /usr/lib32/libgtk-x11-2.0.so”
- /usr/lib/libgtk-x11-2.0.so.0”
- /usr/lib32/libgtk-x11-2.0.so.0”

gdk is loaded with this paths:

- libgdk-x11-2.0.so”
- libgdk-x11-2.0.so.0”
- /usr/lib/libgdk-x11-2.0.so”
- /usr/lib32/libgdk-x11-2.0.so”
- /usr/lib/libgdk-x11-2.0.so.0”
- /usr/lib32/libgdk-x11-2.0.so.0”

For the paths without explicit path, the system will search in /lib, /usr/lib and all directories in the LD_LIBRARY_PATH environment variable.

46.0.232 What does the NAN code mean?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

46.0.233 What font is used as a 'small font' in typical Mac OS X apps?

Plugin Version: all, Platform: macOS.

Answer:

Xojo 4.5 has a constant "SmallSystem" to use for a font name.
For older versions try this code:

Example:

```
Sub GetThemeFont(fontType as Integer, ByRef fontName as String, ByRef fontSize as Integer, ByRef
fontStyle as Integer)
  Var err as Integer
  Var theFont, theFontSize, theFontStyle as MemoryBlock

  const smSystemScript = -1

  Declare Function GetThemeFont Lib "Carbon" (inFontID as Integer, inScript as Integer, outFontName
  as Ptr, outFontSize as Ptr, outStyle as Ptr) as Integer

  theFont = NewMemoryBlock(256) //Str255
  theFontSize = NewMemoryBlock(2) //SInt16
  theFontStyle = NewMemoryBlock(1) //Style

  err = GetThemeFont(fontType, smSystemScript, theFont, theFontSize, theFontStyle)

  if err = 0 then
    fontName = theFont.PString(0)
    fontSize = theFontSize.UShort(0)
    fontStyle = theFontStyle.Byte(0)
  else
    fontName = ""
    fontSize = 0
    fontStyle = 0
  end if
End Sub
```

46.0.234 What is last plugin version to run on Mac OS X 10.4?

Plugin Version: all, Platform: Windows.

Answer: Last Version with 10.4 support is version 15.4.

Notes: With version 15.4 you can build applications for OS X 10.4 and newer.

For Version 16.0 we disabled 10.4 and moved minimum to 10.5. We may be able to enable it again to build a version of 16.x, but may need to charge for this by hour.

46.0.235 What is last plugin version to run on PPC?

Plugin Version: all, Platform: Windows.

Answer: Last Version with PPC is 15.4.

Notes: With version 15.4 you can build PPC applications for OS X 10.4 and newer.

For Version 16.0 we disabled PPC. We may be able to enable it again to build a PPC version of 16.x, but may need to charge for this by hour.

46.0.236 What is last version of the plugins for macOS 32-bit?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use version 23.0 or older.

Notes: We stopped including 32-bit code for macOS in version 23.1.

Please use older versions if you use an old Xojo.

Xojo 2017r3 and newer load our 64-bit plugins.

46.0.237 What is the difference between Timer and WebTimer?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Timer is server side and WebTimer client side.

Notes: Timer is the normal timer class in Xojo. It runs on the server. On the client side the WebTimer runs on the client. It triggers a request to the server to perform the action. So a WebTimer is good to keep the connection running and the website updated regularly. A timer on the server is good to make regular jobs like starting a database backup every 24 hours.

46.0.238 What is the list of Excel functions?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Below a list of function names known by LibXL.

Notes: LibXL parses the functions and writes tokens to the excel file. So even if Excel can do more functions, we can only accept the ones known by LibXL.

ABS, ABSREF, ACOS, ACOSH, ACTIVE.CELL, ADD.BAR, ADD.COMMAND, ADD.MENU, ADD.TOOLBAR, ADDRESS, AND, APP.TITLE, AREAS, ARGUMENT, ASC, ASIN, ASINH, ATAN, ATAN2, ATANH, AVEDEV, AVERAGE, AVERAGEA, BAHTTEXT, BETADIST, BETAINV, BINOMDIST, BREAK, CALL, CALLER, CANCEL.KEY, CEILING, CELL, CHAR, CHECK.COMMAND, CHIDIST, CHIINV, CHITEST,

CHOOSE, CLEAN, CODE, COLUMN, COLUMNS, COMBIN, CONCATENATE, CONFIDENCE, CORREL, COS, COSH, COUNT, COUNTA, COUNTBLANK, COUNTIF, COVAR, CREATE.OBJECT, CRITBINOM, CUSTOM.REPEAT, CUSTOM.UNDO, DATE, DATEDIF, DATESTRING, DATEVALUE, DAVERAGE, DAY, DAYS360, DB, DBCS, DCOUNT, DCOUNTA, DDB, DEGREES, DELETE.BAR, DELETE.COMMAND, DELETE.MENU, DELETE.TOOLBAR, Deref, DEVSQ, DGET, DIALOG.BOX, DIRECTORY, DMAX, DMIN, DOCUMENTS, DOLLAR, DPRODUCT, DSTDEV, DSTDEVP, DSUM, DVAR, DVARP, ECHO, ELSE, ELSE.IF, ENABLE.COMMAND, ENABLE.TOOL, END.IF, ERROR, ERROR.TYPE, EVALUATE, EVEN, EXACT, EXEC, EXECUTE, EXP, EXPONDIST, FACT, FALSE, FCLOSE, FDIST, FILES, FIND, FINDB, FINV, FISHER, FISHERINV, FIXED, FLOOR, FOPEN, FOR, FOR.CELL, FORECAST, FORMULA.CONVERT, FPOS, FREAD, FREADLN, FREQUENCY, FSIZE, FTEST, FV, FWRITE, FWRITELN, GAMMADIST, GAMMAINV, GAMMALN, GEOMEAN, GET.BAR, GET.CELL, GET.CHART.ITEM, GET.DEF, GET.DOCUMENT, GET.FORMULA, GET.LINK.INFO, GET.MOVIE, GET.NAME, GET.NOTE, GET.OBJECT, GET.PIVOT.FIELD, GET.PIVOT.ITEM, GET.PIVOT.TABLE, GET.TOOL, GET.TOOLBAR, GET.WINDOW, GET.WORKBOOK, GET.WORKSPACE, GETPIVOTDATA, GOTO, GROUP, GROWTH, HALT, HARMEAN, HELP, HLOOKUP, HOUR, HYPERLINK, HYPGEOMDIST, IF, INDEX, INDIRECT, INFO, INITIATE, INPUT, INT, INTERCEPT, IPMT, IRR, ISBLANK, ISERR, ISERROR, ISLOGICAL, ISNA, ISNONTEXT, ISNUMBER, ISPMT, ISREF, ISTEXT, ISTHAIDIGIT, KURT, LARGE, LAST.ERROR, LEFT, LEFTB, LEN, LENB, LINEST, LINKS, LN, LOG, LOG10, LOGEST, LOGINV, LOGNORMDIST, LOOKUP, LOWER, MATCH, MAX, MAXA, MDETERM, MEDIAN, MID, MIDB, MIN, MINA, MINUTE, MINVERSE, MIRR, MMULT, MOD, MODE, MONTH, MOVIE.COMMAND, N, NA, NAMES, NEGBINOMDIST, NEXT, NORMDIST, NORMINV, NORMSDIST, NORMSINV, NOT, NOTE, NOW, NPER, NPV, NUMBERSTRING, ODD, OFFSET, OPEN.DIALOG, OPTIONS.LISTS.GET, OR, PAUSE, PEARSON, PERCENTILE, PERCENTRANK, PERMUT, PHONETIC, PI, PIVOT.ADD.DATA, PMT, POISSON, POKE, POWER, PPMT, PRESS.TOOL, PROB, PRODUCT, PROPER, PV, QUARTILE, RADIANS, RAND, RANK, RATE, REFTTEXT, REGISTER, REGISTER.ID, RELREF, RENAME.COMMAND, REPLACE, REPLACEB, REPT, REQUEST, RESET.TOOLBAR, RESTART, RESULT, RESUME, RETURN, RIGHT, RIGHTB, ROMAN, ROUND, ROUNDBAHTDOWN, ROUNDBAHTUP, ROUNDDOWN, ROUNDUP, ROW, ROWS, RSQ, RTD, SAVE.DIALOG, SAVE.TOOLBAR, SCENARIO.GET, SEARCH, SEARCHB, SECOND, SELECTION, SERIES, SET.NAME, SET.VALUE, SHOW.BAR, SIGN, SIN, SINH, SKEW, SLN, SLOPE, SMALL, SPELLING.CHECK, SQRT, STANDARDIZE, STDEV, STDEVA, STDEVP, STDEVPA, STEP, STEYX, SUBSTITUTE, SUBTOTAL, SUM, SUMIF, SUMPRODUCT, SUMSQ, SUMX2MY2, SUMX2PY2, SUMXMY2, SYD, T, TAN, TANH, TDIST, TERMINATE, TEXT, TEXT.BOX, TEXTREF, THAIDAYOFWEEK, THAIDIGIT, THAIMONTHOFYEAR, THAINUMSOUND, THAINUMSTRING, THAISTRINGLENGTH, THAIYEAR, TIME, TIMEVALUE, TINV, TODAY, TRANSPOSE, TREND, TRIM, TRIMMEAN, TRUE, TRUNC, TTEST, TYPE, UNREGISTER, UPPER, USDOLLAR, USERDEFINED, VALUE, VAR, VARA, VARP, VARPA, VDB, VIEW.GET, VLOOKUP, VOLATILE, WEEKDAY, WEIBULL, WHILE, WINDOW.TITLE, WINDOWS, YEAR and ZTEST.

46.0.239 What is the replacement for PluginMBS?

Plugin Version: all, Platform: macOS.

Answer: Use the SoftDeclareMBS class to load libraries dynamically.

46.0.240 What to do on Xojo reporting a conflict?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

I get an error like "This item conflicts with another item of the same name" when using one of the plugin functions.

Xojo just wants to tell you that you dropped something in the plugins folder what is not a plugin.

Notes:

Some users dropped the examples, the documentation or other files into the plugins folder. Don't do it.

46.0.241 What to do with a NSRangeCacheException?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You need to add exception handlers for NSRangeCacheException in order to catch this exception.

Notes: You may also add code to write the stack of the exception into a log file for later locating the error source.

A NSRange has several image representations in memory. So basically you pass in the base image and for whatever size an image is needed, the NSRange class will create a cache image representation of the requested size so on the next query it can use that cache for the same requested size.

46.0.242 What to do with MySQL Error 2014?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can get this error on MySQL if you have a recordset open while you create another one.

46.0.243 What to do with SQL Plugin reporting Malformed string as error?

Plugin Version: all, Platform: macOS.

Answer: Please make sure the table and/or database fields have a text encoding set.

Notes: For Firebird our plugin tries to use UTF-8 encoding if possible and to correctly convert between various tables, the tables and their fields need to have a text encoding defined.

e.g. if the text field in the table is windows-1252 and the other ISO 8859-5, then the Firebird database can convert them to UTF-8 and deliver texts to the plugin.

If encoding is set to none, it may get confused for non-ascii text.

46.0.244 Where is CGGetActiveDisplayListMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetActiveDisplayList.

46.0.245 Where is CGGetDisplaysWithPointMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetDisplaysWithPoint.

46.0.246 Where is CGGetDisplaysWithRectMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetDisplaysWithRect.

46.0.247 Where is CGGetOnlineDisplayListMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetOnlineDisplayList.

46.0.248 Where is GetObjectClassNameMBS?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use this replacement method:

Example:

```
Function GetObjectClassNameMBS(o as Object) As string
  Var t as Introspection.TypeInfo = Introspection.GetType(o)
  Return t.FullName
End Function
```

Notes: GetObjectClassNameMBS was removed from the plugins.

46.0.249 Where is NetworkAvailableMBS?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We removed NetworkAvailableMBS some versions ago. It was not working right and basicly it's not useful. If you want to check whether you have a network, than do a DNS resolve:

Example:

```
// two independend domain names
const domain1 = "www.google.com"
const domain2 = "www.macs.w.de"

// resolve IPs
Var ip1 as string = DNSNameToAddressMBS(Domain1)
Var ip2 as string = DNSNameToAddressMBS(Domain2)

// if we got IPs and not the same IPs (error/login pages)
if len(ip1)=0 or len(ip2)=0 or ip1=ip2 then
MsgBox "no connection"
else
MsgBox "have connection"
end if
```

Notes: This way you can detect whether you got something from DNS. And you can make sure that a DNS redirection to a login page won't catch you.

46.0.250 Where is StringHeight function in DynaPDF?

Plugin Version: all, Platform: Windows.

Answer: Use the function GetFTextHeight or GetFTextHeightEx.

Notes: Be aware that GetFTextHeight works with format commands and you may want to escape your text if you don't use them.

46.0.251 Where is XLSDocumentMBS class?

Plugin Version: all, Platform: macOS.

Answer: This class has been removed in favor of XLBookMBS class.

Notes: This classes have been removed XLSCellMBS, XLSDocumentMBS, XLSFormatRecordMBS, XLSMergedCellsMBS, XLSRowMBS and XLSSheetMBS.

46.0.252 Where to get information about file formats?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

Please visit this web page:
<http://www.wotsit.org>

46.0.253 Where to register creator code for my application?

Plugin Version: all, Platform: macOS.

Answer:

Register at Apple:
<http://developer.apple.com/dev/cftype/information.html>

46.0.254 Which Mac OS X frameworks are 64bit only?

Plugin Version: all, Platform: macOS.

Answer: Some frameworks from Mac OS X do not support 32 bit applications, so we can't provide plugins for Xojo until 64bit target is available.

Notes: For Mac OS X 10.8:

- Accounts
- EventKit
- GLKit
- Social

and in 10.9:

- Accounts
- AVKit
- EventKit
- GameController
- GLKit

- MapKit
- MediaLibrary
- Social
- SpriteKit

In general Apple makes all new frameworks being 64 bit only.

46.0.255 Which plugins are 64bit only?

Plugin Version: all, Platform: macOS.

Answer: Some of our plugins work only in 64 bit modes as operation systems do not provide 32 bit code.

Notes: This effects currently: EventKit, Accounts, Social frameworks from Apple and our matching plugins.

46.0.256 Why application doesn't launch because of a missing ddraw.dll!?

Plugin Version: all, Platform: Windows.

Answer: Some RB versions require that you install DirectX from Microsoft on your Windows.

46.0.257 Why application doesn't launch because of a missing shlwapi.dll!?

Plugin Version: all, Platform: Windows.

Answer: Some RB versions require that you install the Internet Explorer from Microsoft on your Windows.

Notes: This bug is for several older Windows 95 editions.

46.0.258 Why do I hear a beep on keydown?

Plugin Version: all, Platform: Windows.

Answer: When the user presses a key, RB goes through all keydown event handlers till on returns true.

Notes: If no keydown event handler returns true for the key, a beep is performed.

46.0.259 Why does folderitem.item return nil?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Because Xojo fails to make a folderitem for you. Reason may be an alias file which can't be resolved or simply that you don't have enough access rights to read the folder content.

Notes: A more rarely reason is that the directory changed and the file with the given index or name does no longer exist.

46.0.260 Why doesn't showurl work?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

There are three main reasons:

1. showurl is not supported by Xojo in 68k applications.
2. there is now application defined for the protocol (e.g. http) in the Internet Control panel.
3. You don't have Internet Config installed.

You can use the InternetConfigMBS class to check for this stuff.

46.0.261 Why don't the picture functions not work on Linux?

Plugin Version: all, Platform: macOS.

Answer: Please make sure libcairo is installed.

Notes: For accessing pictures on Linux, the MBS Plugin relays on the cairo library. Please install the package if you don't have it already.

Our plugin looks for library called libcairo.so or libcairo.so.2.

46.0.262 Why have I no values in my chart?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You have no data points visible, there may be several reasons:

Notes: For example one of the data values may be infinite or invalid.

Or the scaling may be out of range, so you simply see nothing.

46.0.263 Will application size increase with using plugins?

Plugin Version: all, Platform: Windows.

Answer: All plugins used by your application will be included in the application.

Notes: If you use no plugins, your application will not change size.

And if you use one class from the plugins, your application size will increase by a few kilobytes.

The documentation of the plugins include a list of all plugin parts and their sizes for the different platforms.

46.0.264 XLS: Custom format string guidelines

Plugin Version: all, Platform: macOS.

Answer: You have to download the source code and compile a static version of the library.

Notes: Up to four sections of format codes can be specified. The format codes, separated by semicolons, define the formats for positive numbers, negative numbers, zero values, and text, in that order. If only two sections are specified, the first is used for positive numbers and zeros, and the second is used for negative numbers. If only one section is specified, it is used for all numbers. Four sections example:

```
#,###.00_); [ Red ] (#,###.00);0.00;"sales "@
```

The following table describes the different symbols that are available for use in custom number formats.

Specify colors

To set the text color for a section of the format, type the name of one of the following eight colors in square brackets in the section. The color code must be the first item in the section.

Instead of using the name of the color, the color index can be used, like this [Color3] for Red. Valid numeric indexes for color range from 1 to 56, which reference by index to the legacy color palette.

Specify conditions

To set number formats that will be applied only if a number meets a specified condition, enclose the condition in square brackets. The condition consists of a comparison operator and a value. Comparison operators include: = Equal to; >Greater than; <Less than; >= Greater than or equal to, <= Less than or equal to, and <>Not equal to. For example, the following format displays numbers that are less than or equal to 100 in a red font and numbers that are greater than 100 in a blue font.

```
[ Red ] [ <=100 ] ; [ Blue ] [ >100 ]
```

If the cell value does not meet any of the criteria, then pound signs ("##") are displayed across the width of the cell.

Dates and times

Examples

46.0.265 Xojo doesn't work with your plugins on Windows 98.

Plugin Version: all, Platform: Windows.

Answer: Please upgrade your Windows version.

**46.0.266 Xojo or my RB application itself crashes on launch on Mac OS Classic.
Why?**

Plugin Version: all.

Answer:

You may check if the application has enough memory to be loaded.

RB should have on Mac OS Classic more than 20 MB of RAM.

I preferred to use 50 MB and for an application a 10 MB partition is a good way to start.

Parameter	Description
x	The x value of the data point. For an enumerated x-axis (see <code>Axis.setLabels</code> on what is an enumerated axis), the first data point is 0, and the nth data point is (n-1).
xLabel	The bottom x-axis label of the data point.
x2Label	The top x-axis label of the data point.
value	The value of the data point.
accValue	The sum of values of all data points that are in the same x position and same data group as the current data point, and with data set number less than or equal to the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
totalValue	The sum of values of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
percent	The percentage of the data point based on the total value of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
accPercent	The accumulated percentage of the data point based on the total value of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
gpercent	The percentage of the data point based on the total value of all data points in a layer.
dataSet	The data set number to which the data point belongs. The first data set is 0. The nth data set is (n-1).
dataSetName	The name of the data set to which the data point belongs.
dataItem	The data point number within the data set. The first data point is 0. The nth data point is (n-1).
dataGroup	The data group number to which the data point belongs. The first data group is 0. The nth data group is (n-1).
dataGroupName	The name of the data group to which the data point belongs.
layerId	The layer number to which the data point belongs. The first layer is 0. The nth layer is (n-1).
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using <code>Layer.addExtraField</code> , <code>Layer.addExtraField2</code> , <code>BaseChart.addExtraField</code> or <code>BaseChart.addExtraField2</code> .

diFieldN	Same as fieldN. See above.
dsFieldN	Similar to fieldN, except that dsFieldN means the extra field is indexed by data set number. The Pth data set corresponds to the Pth element of the extra field.
dsdiFieldN	Similar to fieldN, except that dsdiFieldN means the extra fields are indexed by both the data set number and data point number. The Pth data item of the Qth data set corresponds to the Pth element of the (N + Q)th extra field.

Parameter	Description
zx	The symbol scale in the x dimension. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .
zy	The symbol scale in the y dimension. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .
z	The symbol scale without distinguishing the dimension to use. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .

Parameter	Description
slope	The slope of the trend line.
intercept	The y-intercept of the trend line.
corr	The correlation coefficient in linear regression analysis.
stderr	The standard error in linear regression analysis.

Parameter	Description
top	The value of the top edge of the box-whisker symbol.
bottom	The value of the bottom edge of the box-whisker symbol.
max	The value of the maximum mark of the box-whisker symbol.
min	The value of the minimum mark of the box-whisker symbol.
med	The value of the median mark of the box-whisker symbol.

Parameter	Description
high	The high value.
low	The low value.
open	The open value.
close	The close value.

Parameter	Description
dir	The direction of the vector.
len	The length of the vector.

Parameter	Description
radius	The radial value of the data point.
value	Same as { radius } . See above.
angle	The angular value of the data point.
x	Same as { angle } . See above.
label	The angular label of the data point.
xLabel	Same as { label } . See above.
name	The name of the layer to which the data point belongs.
dataSetName	Same as { name } . See above.
i	The data point number. The first data point is 0. The nth data point is (n-1).
dataItem	Same as { i } . See above.
z	The symbol scale. Applicable for layers with symbol scales set by Polar-Layer.setSymbolScale.
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using Layer.addExtraField, Layer.addExtraField2, BaseChart.addExtraField or BaseChart.addExtraField2.
diFieldN	Same as fieldN. See above.
dsFieldN	Similar to fieldN, except that dsFieldN means the extra field is indexed by layer index. The Pth layer corresponds to the Pth element of the extra field.
dsdiFieldN	Similar to fieldN, except that dsdiFieldN means the extra fields are indexed by both the data set number and data point number. The Pth data item of the Qth layer corresponds to the Pth element of the (N + Q)th extra field.
Parameter	Description
dir	The direction of the vector.
len	The length of the vector.
Parameter	Description
value	The axis value at the tick position.
label	The axis label at the tick position.
Parameter	Description
[param]	The name of the parameter
[a]	If this field a number, it specifies the number of decimal places (digits to the right of the decimal point).

[b]	The thousand separator. Should be a non-alphanumeric character (not 0-9, A-Z, a-z). Use ' '.
textasciitilde ' for no thousand separator. The default is ' '.	
textasciitilde ', which can be modified using BaseChart.setNumberFormat.	
[c]	The decimal point character. The default is '.', which can be modified using BaseChart.setNumberFormat.
[d]	The negative sign character. Use ' '.
textasciitilde ' for no negative sign character. The default is '-', which can be modified using BaseChart.setNumberFormat.	

Parameter	Description
yyyy	The year in 4 digits (e.g. 2002)
yyy	The year showing only the least significant 3 digits (e.g. 002 for the year 2002)
yy	The year showing only the least significant 2 digits (e.g. 02 for the year 2002)
y	The year showing only the least significant 1 digits (e.g. 2 for the year 2002)
mmm	The month formatted as its name. The default is to use the first 3 characters of the english month name (Jan, Feb, Mar ...). The names can be configured using BaseChart.setMonthNames.
mm	The month formatted as 2 digits from 01 - 12, adding leading zero if necessary.
m	The month formatted using the minimum number of digits from 1 - 12.
MMM	The first 3 characters of the month name converted to upper case. The names can be configured using BaseChart.setMonthNames.
MM	The first 2 characters of the month name converted to upper case. The names can be configured using BaseChart.setMonthNames.
M	The first character of the month name converted to upper case. The names can be configured using BaseChart.setMonthNames.
dd	The day of month formatted as 2 digits from 01 - 31, adding leading zero if necessary.
d	The day of month formatted using the minimum number of digits from 1 - 31.
w	The name of the day of week. The default is to use the first 3 characters of the english day of week name (Sun, Mon, Tue ...). The names can be configured using BaseChart.setWeekDayNames.
hh	The hour of day formatted as 2 digits, adding leading zero if necessary. The 2 digits will be 00 - 23 if the 'a' option (see below) is not specified, otherwise it will be 01 - 12.
h	The hour of day formatted using the minimum number of digits. The digits will be 0 - 23 if the 'a' option (see below) is not specified, otherwise it will be 01 - 12.
nn	The minute formatted as 2 digits from 00 - 59, adding leading zero if necessary.
n	The minute formatted using the minimum number of digits from 00 - 59.
ss	The second formatted as 2 digits from 00 - 59, adding leading zero if necessary.
s	The second formatted using the minimum number of digits from 00 - 59.
a	Display either 'am' or 'pm', depending on whether the time is in the morning or afternoon. The text 'am' and 'pm' can be modified using BaseChart.setAMPM.

Shape Id	Value	Description
SquareShape	1	Square shape. See (1, 1) above.
DiamondShape	2	Diamond shape. See (2, 1) above.
TriangleShape	3	Triangle shape pointing upwards. See (3, 1) above.
RightTriangleShape	4	Triangle shape pointing rightwards. See (4, 1) above.
LeftTriangleShape	5	Triangle shape pointing leftwards. See (5, 1) above.
InvertedTriangleShape	6	Triangle shape pointing downwards. See (1, 2) above.
CircleShape	7	Circle shape. See (2, 2) above.
StarShape	[Method]	Star shapes of various points. See (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5) above for stars with 3 to 10 points.
PolygonShape	[Method]	Polygon shapes symmetrical about a vertical axis with a vertex at the top center position. See (4, 1), (4, 3), (4, 5), (5, 1) for polygons of 5 to 8 sides.
Polygon2Shape	[Method]	Polygon shapes symmetrical about a vertical axis but without any vertex at the top center position. See (4, 2), (4, 4) for polygons of 5 and 6 sides.
CrossShape	[Method]	'+' shapes. See (5, 2), (5, 3), (5, 4), (5, 5), (6, 1), (6, 2), (6, 3) for '+' shape with arm width of 0.1 - 0.7.
Cross2Shape	[Method]	'X' shapes. See (6, 4), (6, 5), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5) for 'X' shapes with arm width of 0.1 - 0.7.

langEnglish	0	Roman script
langFrench	1	Roman script
langGerman	2	Roman script
langItalian	3	Roman script
langDutch	4	Roman script
langSwedish	5	Roman script
langSpanish	6	Roman script
langDanish	7	Roman script
langPortuguese	8	Roman script
langNorwegian	9	Roman script
langHebrew	10	Hebrew script
langJapanese	11	Japanese script
langArabic	12	Arabic script
langFinnish	13	Roman script
langGreek	14	Greek script using smRoman script code
langIcelandic	15	modified smRoman/Icelandic script
langMaltese	16	Roman script
langTurkish	17	modified smRoman/Turkish script
langCroatian	18	modified smRoman/Croatian script
langTradChinese	19	Chinese (Mandarin) in traditional characters
langUrdu	20	Arabic script
langHindi	21	Devanagari script
langThai	22	Thai script
langKorean	23	Korean script

Nan	Meaning
1	Invalid square root (negative number, usually)
2	Invalid addition (indeterminate such as infinity + (-infinity))
4	Invalid division (indeterminate such as 0/0)
8	Invalid multiplication (indeterminate such as 0*infinity)
9	Invalid modulo such as (a mod 0)
17	Try to convert invalid string to a number like val("x7")
33	Invalid argument in a trig function
34	Invalid argument in an inverse trig function
36	Invalid argument in a log function
37	Invalid argument in Pow function
38	Invalid argument in toolbox financial function
40	Invalid argument in hyperbolic function
42	Invalid argument in a gamma function

Symbol	Description and result
0	Digit placeholder. For example, if the value 8.9 is to be displayed as 8.90, use the format #.00
#	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall not display extra zeros when the number typed has fewer digits on either side of the decimal than there are # symbols in the format. For example, if the custom format is #.##, and 8.9 is in the cell, the number 8.9 is displayed.
?	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall put a space for insignificant zeros on either side of the decimal point so that decimal points are aligned in the column. For example, the custom format 0.0? aligns the decimal points for the numbers 8.9 and 88.99 in a column.
. (period)	Decimal point.
%	Percentage. If the cell contains a number between 0 and 1, and the custom format 0% is used, the application shall multiply the number by 100 and add the percentage symbol in the cell.
, (comma)	Thousands separator. The application shall separate thousands by commas if the format contains a comma that is enclosed by number signs (#) or by zeros. A comma that follows a placeholder scales the number by one thousand. For example, if the format is #.0,, and the cell value is 12,200,000 then the number 12.2 is displayed.
E- E+ e- e+	Scientific format. The application shall display a number to the right of the "E" symbol that corresponds to the number of places that the decimal point was moved. For example, if the format is 0.00E+00, and the value 12,200,000 is in the cell, the number 1.22E+07 is displayed. If the number format is #0.0E+0, then the number 12.2E+6 is displayed.
\$ -+/():space	Displays the symbol. If it is desired to display a character that differs from one of these symbols, precede the character with a backslash (\). Alternatively, enclose the character in quotation marks. For example, if the number format is (000), and the value 12 is in the cell, the number (012) is displayed.
\	Display the next character in the format. The application shall not display the backslash. For example, if the number format is 0\!, and the value 3 is in the cell, the value 3! is displayed.
*	Repeat the next character in the format enough times to fill the column to its current width. There shall not be more than one asterisk in one section of the format. If more than one asterisk appears in one section of the format, all but the last asterisk shall be ignored. For example, if the number format is 0*x, and the value 3 is in the cell, the value 3xxxxxx is displayed. The number of x characters that are displayed in the cell varies based on the width of the column.
_ (underline)	Skip the width of the next character. This is useful for lining up negative and positive values in different cells of the same column. For example, the number format _(0.0_);(0.0) aligns the numbers 2.3 and -4.5 in the column even though the negative number is enclosed by parentheses.
"text"	Display whatever text is inside the quotation marks. For example, the format 0.00 "dollars" displays 1.23 dollars when the value 1.23 is in the cell.
@	Text placeholder. If text is typed in the cell, the text from the cell is placed in the format where the at symbol (@) appears. For example, if the number format is "Bob "@ Smith" (including quotation marks), and the value "John" is in the cell, the value Bob John Smith is displayed.

[Black] [Green] [White] [Blue] [Magenta] [Yellow] [Cyan] [Red]

To display	As	Use this code
Months	1-12	m
Months	01-12	mm
Months	Jan-Dec	mmm
Months	January-December	mmmm
Months	J-D	mmmmm
Days	1-31	d
Days	01-31	dd
Days	Sun-Sat	ddd
Days	Sunday-Saturday	dddd
Years	00-99	yy
Years	1900-9999	yyyy
Hours	0-23	h
Hours	00-23	hh
Minutes	0-59	m
Minutes	00-59	mm
Seconds	0-59	s
Seconds	00-59	ss
Time	4 AM	h AM/PM
Time	4:36 PM	h:mm AM/PM
Time	4:36:03 P	h:mm:ss A/P
Time	4:36:03.75	h:mm:ss.00
Elapsed time	1:02	[h] :mm
Elapsed time	62:16	[mm] :ss
Elapsed time	3735.80	[ss] .00

To display	As	Use this code
1234.59	1234.6	#####.#
8.9	8.900	#.000
.631	0.6	0.#
12	12.0	#.0#
1234.568	1234.57	#.0#
44.398	44.398	???.???
102.65	102.65	???.???
2.8	2.8	???.???
5.25	5 1/4	# ??/??
5.3	5 3/10	# ??/??
12000	12,000	#,###
12000	12	#,
12400000	12.4	0.0,,