MBS RabbitMQ Plugin Documentation

Christian Schmitz

November 17, 2025

0.1 Introduction

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

0.2 Content

- 1 List of all topics 3
- 2 List of all classes 9
- 3 All items in this plugin 11

Chapter 1

List of Topics

- 3 **RabbitMQ** 11
  - 3.1.1 class RabbitMQBasicPropertiesMBS 11
    - 3.1.3 Constructor 11
    - 3.1.4 Destructor 11
    - 3.1.6 AppId as String 12
    - 3.1.7 ClusterId as String 12
    - 3.1.8 ContentEncoding as String 12
    - 3.1.9 ContentType as String 12
    - 3.1.10 CorrelationId as String 12
    - 3.1.11 DeliveryMode as Integer 13
    - 3.1.12 Expiration as String 13
    - 3.1.13 Flags as Integer 13
    - 3.1.14 HasAppId as Boolean 13
    - 3.1.15 HasClusterId as Boolean 13
    - 3.1.16 HasContentEncoding as Boolean 13
    - 3.1.17 HasContentType as Boolean 14
    - 3.1.18 HasCorrelationId as Boolean 14
    - 3.1.19 HasDeliveryMode as Boolean 14
    - 3.1.20 HasExpiration as Boolean 14
    - 3.1.21 HasHeaders as Boolean 14
    - 3.1.22 HasMessageId as Boolean 15
    - 3.1.23 HasPriority as Boolean 15
    - 3.1.24 HasReplyTo as Boolean 15
    - 3.1.25 HasTimestamp as Boolean 15
    - 3.1.26 HasType as Boolean 15
    - 3.1.27 HasUserId as Boolean 15
    - 3.1.28 Headers as Dictionary 16

* 3.1.29 MessageId as String 16
* 3.1.30 Priority as Integer 16
* 3.1.31 ReplyTo as String 16
* 3.1.32 Timestamp as UInt64 16
* 3.1.33 Type as String 17
* 3.1.34 UserId as String 17

– 3.2.1 class RabbitMQConnectionInfoMBS 18

* 3.2.3 Constructor 18
* 3.2.4 Constructor(URL as String) 18
* 3.2.5 Destructor 19
* 3.2.7 Host as String 19
* 3.2.8 Password as String 19
* 3.2.9 Port as Integer 19
* 3.2.10 SSL as Boolean 20
* 3.2.11 User as String 20
* 3.2.12 VHost as String 20

– 3.3.1 class RabbitMQConnectionMBS 21

* 3.3.3 BasicAck(channel as Integer, deliveryTag as UInt64, multiple as Boolean = false) as Integer 21
* 3.3.4 BasicCancel(channel as Integer, consumerTag as String) as Dictionary 21
* 3.3.5 BasicConsume(channel as Integer, queue as String, consumerTag as String, noLocal as Boolean, noAck as Boolean, exclusive as Boolean, arguments as Dictionary) as Dictionary 22
* 3.3.6 BasicGet(channel as Integer, queue as String, noAck as Boolean) as RabbitMQRPCReplyMBS 22
* 3.3.7 BasicNAck(channel as Integer, deliveryTag as UInt64, multiple as Boolean = false, requeue as Boolean = false) as Integer 22
* 3.3.8 BasicPublish(channel as Integer, exchange as String, routingKey as String, mandatory as Boolean, immediate as Boolean, properties as RabbitMQBasicPropertiesMBS, body as String) as Integer 23
* 3.3.9 BasicQOS(channel as Integer, prefetchSize as Integer, prefetchCount as Integer, isGlobal as boolean) as Dictionary 24
* 3.3.10 BasicRecover(channel as Integer, requeue as Boolean) as Dictionary 24
* 3.3.11 BasicReject(channel as Integer, deliveryTag as UInt64, requeue as Boolean = false) as Integer 24
* 3.3.12 ChannelClose(Channel as Integer, Code as Integer = 200) as RabbitMQRPCReplyMBS 24
* 3.3.13 ChannelFlow(channel as Integer, active as Boolean) as Dictionary 25
* 3.3.14 ChannelOpen(Channel as Integer) as Dictionary 25
* 3.3.15 ConfirmSelect(channel as Integer) as Dictionary 25
* 3.3.16 ConnectionClose(Code as Integer = 200) as RabbitMQRPCReplyMBS 25
* 3.3.17 ConstantIsHardError(constantNumber as Integer) as Boolean 25
* 3.3.18 ConstantName(constantNumber as Integer) as String 25

* 3.3.19 Constructor 26
* 3.3.20 ConsumeMessage(byref envelope as RabbitMQEnvelopeMBS, timeoutValue as double = 30, flags as Integer = 0) as RabbitMQRPCReplyMBS 26
* 3.3.21 Destructor 26
* 3.3.22 ErrorString(ErrorCode as Integer) as String 27
* 3.3.23 ExchangeBind(channel as Integer, destination as String, source as String, routingKey as String, arguments as Dictionary) as Dictionary 27
* 3.3.24 ExchangeDeclare(channel as Integer, exchange as String, type as String, passive as Boolean, durable as Boolean, autoDelete as Boolean, internal as Boolean, arguments as Dictionary) as Dictionary 27
* 3.3.25 ExchangeDelete(channel as Integer, exchange as String, ifUsed as Boolean) as Dictionary 27
* 3.3.26 ExchangeUnbind(channel as Integer, destination as String, source as String, routingKey as String, arguments as Dictionary) as Dictionary 27
* 3.3.27 LoginExternal(vhost as String, channelMax as Integer, frameMax as Integer, heartbeat as Integer, Identity as String, properties as Dictionary = nil) as RabbitMQRPCReplyMBS 28
* 3.3.28 LoginPlain(vhost as String, channelMax as Integer, frameMax as Integer, heartbeat as Integer, Username as String, Password as String, properties as Dictionary = nil) as RabbitMQRPCReplyMBS 29
* 3.3.29 MaybeReleaseBuffers 29
* 3.3.30 MaybeReleaseBuffersOnChannel(channel as Integer) 30
* 3.3.31 MethodHasContent(constantNumber as Integer) as Boolean 30
* 3.3.32 MethodName(constantNumber as Integer) as String 30
* 3.3.33 NewTCPSocket(UseSSL as boolean = false) as Boolean 30
* 3.3.34 OpenSocket(Host as String, Port as Integer, TimeOut as Double = 10) as Integer 31
* 3.3.35 QueueBind(channel as Integer, queue as String, exchange as String, routingKey as String, arguments as Dictionary) as Dictionary 31
* 3.3.36 QueueDeclare(channel as Integer, queue as String, passive as Boolean, durable as Boolean, exclusive as Boolean, autoDelete as Boolean, arguments as Dictionary) as Dictionary 31
* 3.3.37 QueueDelete(channel as Integer, queue as String, ifUnused as Boolean, ifEmpty as Boolean) as Dictionary 31
* 3.3.38 QueuePurge(channel as Integer, queue as String) as Dictionary 32
* 3.3.39 QueueUnbind(channel as Integer, queue as String, exchange as String, routingKey as String, arguments as Dictionary) as Dictionary 32
* 3.3.40 ReadMessage(channel as Integer, byref message as RabbitMQMessageMBS, flags as Integer = 0) as RabbitMQRPCReplyMBS 32
* 3.3.41 SendHeader as Integer 32
* 3.3.42 SetCACert(File as FolderItem) as Boolean 33
* 3.3.43 SetCACert(Path as String) as Boolean 33
* 3.3.44 SetKey(CertFile as FolderItem, KeyFile as FolderItem) as Boolean 33
* 3.3.45 SetKey(CertPath as String, KeyPath as String) as Boolean 33
* 3.3.46 SetKeyData(CertFile as FolderItem, KeyData as String) as Boolean 34

* 3.3.47 SetKeyData(CertPath as String, KeyData as String) as Boolean 34
* 3.3.48 SimpleWaitFrame(byref Frame as RabbitMQFrameMBS, timeoutValue as double = 30) as Integer 34
* 3.3.49 TXCommit(channel as Integer) as Dictionary 35
* 3.3.50 TXRollback(channel as Integer) as Dictionary 35
* 3.3.51 TXSelect(channel as Integer) as Dictionary 35
* 3.3.53 ClientProperties as Dictionary 36
* 3.3.54 DataInBuffer as Boolean 36
* 3.3.55 FramesEnqueued as Boolean 36
* 3.3.56 Handle as Integer 36
* 3.3.57 HandshakeTimeout as Double 37
* 3.3.58 Heartbeat as Integer 37
* 3.3.59 MaximumChannel as Integer 37
* 3.3.60 MaximumFrameSize as Integer 37
* 3.3.61 RPCReply as RabbitMQRPCReplyMBS 38
* 3.3.62 RPCTimeout as Double 39
* 3.3.63 ServerProperties as Dictionary 39
* 3.3.64 SocketFileDescriptor as Integer 39
* 3.3.65 SSLCACertPath as String 40
* 3.3.66 SSLCertPath as String 40
* 3.3.67 SSLKeyData as String 40
* 3.3.68 SSLKeyPath as String 40
* 3.3.69 SSLMaxVersion as Integer 40
* 3.3.70 SSLMinVersion as Integer 41
* 3.3.71 SSLVerifyHostname as Boolean 41
* 3.3.72 SSLVerifyPeer as Boolean 41
* 3.3.73 Version as String 41
* 3.3.74 VersionNumber as Integer 41

– 3.4.1 class RabbitMQEnvelopeMBS 44

* 3.4.3 Constructor 44
* 3.4.4 Destructor 44
* 3.4.6 Channel as Integer 44
* 3.4.7 ConsumerTag as String 44
* 3.4.8 DeliveryTag as UInt64 45
* 3.4.9 Exchange as String 45
* 3.4.10 Message as RabbitMQMessageMBS 45
* 3.4.11 Redelivered as Boolean 45
* 3.4.12 RoutingKey as String 45

– 3.5.1 class RabbitMQFrameMBS 46

* 3.5.3 Constructor 46
* 3.5.4 Destructor 46

* 3.5.6 BodyFragment as MemoryBlock 46
* 3.5.7 BodyFragmentSize as Integer 46
* 3.5.8 BodySize as Integer 47
* 3.5.9 Channel as Integer 47
* 3.5.10 ClassID as Integer 47
* 3.5.11 FrameType as Integer 47
* 3.5.12 MethodDecoded as Dictionary 47
* 3.5.13 MethodID as Integer 48
* 3.5.14 PropertiesDecoded as Dictionary 48
* 3.5.15 Raw as MemoryBlock 48
* 3.5.16 RawSize as Integer 48

– 3.6.1 class RabbitMQMessageMBS 52

* 3.6.3 Constructor 52
* 3.6.5 Body as MemoryBlock 52
* 3.6.6 Properties as RabbitMQBasicPropertiesMBS 52

– 3.7.1 class RabbitMQRPCReplyMBS 54

* 3.7.3 Constructor 54
* 3.7.4 Destructor 54
* 3.7.6 LibraryError as Integer 54
* 3.7.7 MethodDecoded as Dictionary 55
* 3.7.8 MethodId as Integer 55
* 3.7.9 Type as Integer 55

Chapter 2

List of all classes

- RabbitMQBasicPropertiesMBS 11
- RabbitMQConnectionInfoMBS 18
- RabbitMQConnectionMBS 21
- RabbitMQEnvelopeMBS 44
- RabbitMQFrameMBS 46
- RabbitMQMessageMBS 52
- RabbitMQRPCReplyMBS 54

Chapter 3

RabbitMQ

3.1 class RabbitMQBasicPropertiesMBS

3.1.1 class RabbitMQBasicPropertiesMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The basic properties class.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.1.2 Methods

3.1.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The constructor.

3.1.4 Destructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.

3.1.5 Properties

3.1.6 AppId as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The app-id.
Notes: (Read and Write property)

3.1.7 ClusterId as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The cluster ID.
Notes: (Read and Write property)

3.1.8 ContentEncoding as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The content-encoding.
Notes: (Read and Write property)

3.1.9 ContentType as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The content-type.
Notes: (Read and Write property)

3.1.10 CorrelationId as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The correlation-id.
Notes: (Read and Write property)

3.1.11 DeliveryMode as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The delivery-mode.
Notes: (Read and Write property)

3.1.12 Expiration as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The expiration.
Notes: (Read and Write property)

3.1.13 Flags as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Bit-mask of set fields.
Notes: (Read and Write property)

3.1.14 HasAppId as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagAppId.
Notes: (Read and Write property)

3.1.15 HasClusterId as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagClusterId.
Notes: (Read and Write property)

3.1.16 HasContentEncoding as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagContentEncoding.
Notes: (Read and Write property)

3.1.17 HasContentType as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagContentType.
Notes: (Read and Write property)

3.1.18 HasCorrelationId as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagCorrelationId.
Notes: (Read and Write property)

3.1.19 HasDeliveryMode as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagDeliveryMode.
Notes: (Read and Write property)

3.1.20 HasExpiration as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagExpiration.
Notes: (Read and Write property)

3.1.21 HasHeaders as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagHeaders.
Notes: (Read and Write property)

3.1.22 HasMessageId as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagMessageId.
Notes: (Read and Write property)

3.1.23 HasPriority as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagPriority.
Notes: (Read and Write property)

3.1.24 HasReplyTo as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagReplyTo.
Notes: (Read and Write property)

3.1.25 HasTimestamp as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagTimestamp.
Notes: (Read and Write property)

3.1.26 HasType as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagType.
Notes: (Read and Write property)

3.1.27 HasUserId as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Convenience property to set/get Flags with kFlagUserId.
Notes: (Read and Write property)

3.1.28 Headers as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Dictionary with headers.
Notes: (Read and Write property)

3.1.29 MessageId as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The message ID.
Notes: (Read and Write property)

3.1.30 Priority as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The priority.
Notes: (Read and Write property)

3.1.31 ReplyTo as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The reply to field.
Notes: (Read and Write property)

3.1.32 Timestamp as UInt64

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The timestamp.
Notes: (Read and Write property)

3.1.33 Type as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The type field.
Notes: (Read and Write property)

3.1.34 UserId as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The user id.
Notes: (Read and Write property)

3.1.35 Constants

Delivery modes

Constant	Value	Description
kDeliveryNonPersistent	1	Non-persistent message.
kDeliveryPersistent	2	Persistent message

Field flags

Constant	Value	Description
kFlagAppId	8	AppID field is set.
kFlagClusterId	4	ClusterId field is set.
kFlagContentEncoding	16384	ContentEncoding field is set.
kFlagContentType	32768	ContentType field is set.
kFlagCorrelationId	1024	CorrelationId field is set.
kFlagDeliveryMode	4096	DeliveryMode field is set.
kFlagExpiration	256	Expiration field is set.
kFlagHeaders	8192	Headers field is set.
kFlagMessageId	128	MessageId field is set.
kFlagPriority	2048	Priority field is set.
kFlagReplyTo	512	ReplyTo field is set.
kFlagTimestamp	64	Timestamp field is set.
kFlagType	32	Type field is set.
kFlagUserId	16	UserId field is set.

3.2 class RabbitMQConnectionInfoMBS

3.2.1 class RabbitMQConnectionInfoMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class ot parse an URL.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.2.2 Methods

3.2.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The constructor.
Notes: Sets default values for properties.

The default values are:

- user: "guest"
- password: "guest"
- host: "localhost"
- vhost: "/"
- port: 5672

See also:

- 3.2.4 Constructor(URL as String) 18

3.2.4 Constructor(URL as String)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Parse a connection URL.
Notes: An amqp connection url takes the form:

amqp:// [$ USERNAME [:$ PASSWORD] @] $ HOST [:$ PORT] / [$ VHOST]

Examples:
amqp://guest:guest@localhost:5672//
amqp://guest:guest@localhost/myvhost

Any missing parts of the URL will be set to the defaults specified in other Constructor. For amqps: URLs the default port will be set to 5671 instead of 5672 for non-SSL URLs.

We raise UnsupportedOperationException if parsing failes.
See also:

- 3.2.3 Constructor 18

3.2.5 Destructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.

3.2.6 Properties

3.2.7 Host as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The hostname of the broker.
Notes: (Read only property)

3.2.8 Password as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The password to authenticate with the broker.
Notes: default on most brokers is 'guest'
(Read only property)

3.2.9 Port as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The port that the broker is listening on.
Notes: default on most brokers is 5672.
(Read only property)

3.2.10 SSL as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Whether to use SSL.
Notes: (Read only property)

3.2.11 User as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The username to authenticate with the broker.
Notes: default on most broker is 'guest'
(Read only property)

3.2.12 VHost as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The virtual host on the broker to connect to.
Notes: a good default is "/"
(Read only property)

3.3 class RabbitMQConnectionMBS

3.3.1 class RabbitMQConnectionMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for a RabbitMQ connection to a server.
Notes: Our plugin converts from dictionary/array/variant to/from the encoded representation used in MQ automatically.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.3.2 Methods

3.3.3 BasicAck(channel as Integer, deliveryTag as UInt64, multiple as Boolean = false) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Acknowledges a message.
Notes: Does a basic.ack on a received message.

channel: the channel identifier
deliveryTag: the delivery tag of the message to be ack'd.
multiple: if true, ack all messages up to this delivery tag, if false ack only this delivery tag.

Returns error code, 0 on success, 0 >on failing to send the ack to the broker. this will not indicate failure if something goes wrong on the broker.

3.3.4 BasicCancel(channel as Integer, consumerTag as String) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Cancels a channel.

3.3.5 BasicConsume(channel as Integer, queue as String, consumerTag as String, noLocal as Boolean, noAck as Boolean, exclusive as Boolean, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Basic consume.

3.3.6 BasicGet(channel as Integer, queue as String, noAck as Boolean) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Do a basic.get.
Notes: Synchronously polls the broker for a message in a queue, and retrieves the message if a message is in the queue.

channel: the channel identifier to use
queue: the queue name to retrieve from
noAck: if true the message is automatically ack'ed if false BasicAck() should be called once the message retrieved has been processed

Returns RabbitMQRPCReplyMBS object indicating success or failure

3.3.7 BasicNAck(channel as Integer, deliveryTag as UInt64, multiple as Boolean = false, requeue as Boolean = false) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Do a basic.nack.
Notes: Actively reject a message, this has the same effect as BasicReject() however, BasicNAck() can negatively acknowledge multiple messages with one call much like BasicAck() can acknowledge mutliple messages with
one call.

channel: the channel identifier
deliveryTag the delivery tag of the message to reject
multiple: if set to 1 negatively acknowledge all unacknowledged messages on this channel.
requeue: indicate to the broker whether it should requeue the message or dead-letter it.

Returns error code, kStatusOK on success, an status constant value otherwise.

3.3.8 BasicPublish(channel as Integer, exchange as String, routingKey as String, mandatory as Boolean, immediate as Boolean, properties as RabbitMQBasicPropertiesMBS, body as String) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Publish a message to the broker.
Notes: Publish a message on an exchange with a routing key.

Note that at the AMQ protocol level basic.publish is an async method:
this means error conditions that occur on the broker (such as publishing to
a non-existent exchange) will not be reflected in the return value of this
function.

channel: the channel identifier
exchange: the exchange on the broker to publish to
routingKey: the routing key to use when publishing the message mandatory indicate to the broker that the message MUST be routed to a queue. If the broker cannot do this it should respond with a basic.return method.
immediate: indicate to the broker that the message MUST be delivered to a consumer immediately. If the broker cannot do this it should respond with a basic.return method.
properties: the properties associated with the message
body: the message body

kStatusOK on success, other kStatus values on failure. Note that basic.publish is an async method, the return value from this function only indicates that the message data was successfully transmitted to the broker. It does not indicate failures that occur on the broker, such as publishing to a non-existent exchange.

Possible error values:

- kStatusTimerFailure: system timer facility returned an error the message was not sent.
- kStatusHeartbeatTimeout: connection timed out waiting for a heartbeat from the broker. The message was not sent.
- kStatusNoMemory: memory allocation failed. The message was not sent.
- kStatusTableTooBig: a table in the properties was too large to fit in a single frame. Message was not sent.
- kStatusConnectionClosed: the connection was closed
- kStatusSSLError: a SSL error occurred.
- kStatusTCPError: a TCP error occurred. errno or WSAGetLastError() may provide more information.

3.3.9 BasicQOS(channel as Integer, prefetchSize as Integer, prefetchCount as Integer, isGlobal as boolean) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Basic QOS.

3.3.10 BasicRecover(channel as Integer, requeue as Boolean) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Recover a channel.

3.3.11 BasicReject(channel as Integer, deliveryTag as UInt64, requeue as Boolean = false) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Do a basic.reject.
Notes: Actively reject a message that has been delivered

channel: the channel identifier
deliveryTag the delivery tag of the message to reject
requeue: indicate to the broker whether it should requeue the message or just discard it.

Returns error code, 0 on success, 0 >on failing to send the reject method to the broker.
This will not indicate failure if something goes wrong on the broker.

3.3.12 ChannelClose(Channel as Integer, Code as Integer = 200) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Closes an channel.
Notes: channel: the channel identifier
code: the reason for closing the channel, kReplySuccess is a good default.

Returns RabbitMQRPCReplyMBS object indicating success or failure

3.3.13 ChannelFlow(channel as Integer, active as Boolean) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Channel Flow control.

3.3.14 ChannelOpen(Channel as Integer) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Opens a channel.

3.3.15 ConfirmSelect(channel as Integer) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Confirm select.

3.3.16 ConnectionClose(Code as Integer = 200) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Closes a connection.

3.3.17 ConstantIsHardError(constantNumber as Integer) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks to see if a constant is a hard error.
Notes: A hard error occurs when something severe enough happens that the connection must be closed.

Returns true if its a hard error, false otherwise.

3.3.18 ConstantName(constantNumber as Integer) as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get constant name string from constant.

3.3.19 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The constructor.
Notes: Allocate and initialize a new connection object.

3.3.20 ConsumeMessage(byref envelope as RabbitMQEnvelopeMBS, timeoutValue as double = 30, flags as Integer = 0) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Wait for and consume a message.
Notes: Waits for a basic.deliver method on any channel, upon receipt of basic.deliver it reads that message, and returns. If any other method is received before basic.deliver, this function will return an RabbitMQRPCReplyMBS with ReplyType = ResponseTypeLibraryException, and Reply.LibraryError = kStatusUnexpectedState. The caller should then call WaitFrame() to read this frame and take appropriate action.

This function should be used after starting a consumer with the BasicConsume() function

Envelope receives new object.
Timeout is in seconds.
flags: pass in 0. Currently unused.

Returns RabbitMQRPCReplyMBS, which may contain error information.
Type of the object is RabbitMQRPCReplyMBS.ResponseTypeNormal on success.

3.3.21 Destructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.
Notes: Closes the entire connection.

Implicitly closes all channels and informs the broker the connection
is being closed, after receiving acknowledgment from the broker it closes
the socket.

3.3.22 ErrorString(ErrorCode as Integer) as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the error string for the given error code.
Notes: Get an error string associated with an error code.

3.3.23 ExchangeBind(channel as Integer, destination as String, source as String, routingKey as String, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Binds an exchange.

3.3.24 ExchangeDeclare(channel as Integer, exchange as String, type as String, passive as Boolean, durable as Boolean, autoDelete as Boolean, internal as Boolean, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Declare an exchange.

3.3.25 ExchangeDelete(channel as Integer, exchange as String, ifUsed as Boolean) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Deletes an exchange.

3.3.26 ExchangeUnbind(channel as Integer, destination as String, source as String, routingKey as String, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Unbinds an exchange.

3.3.27 LoginExternal(vhost as String, channelMax as Integer, frameMax as Integer, heartbeat as Integer, Identity as String, properties as Dictionary = nil) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Login with external SASL.
Notes: Login to the broker passing an optional properties table.

vhost: the virtual host to connect to on the broker. The default on most brokers is "/".
channelMax: the limit for the number of channels for the connection.
0 means no limit, and is a good default (kDefaultMaxChannels) Note that the maximum number of channels the protocol supports is 65535 (2^16, with the 0-channel reserved). The server can set a lower channelMax and then the client will use the lowest of the two.
frameMax: the maximum size of an AMQP frame ont he wire to request of the broker for this connection. 4096 is the minimum size, 2^31-1 is the maximum, a good default is 131072 (128KB), or kDefaultFrameSize.
heartbeat: the number of seconds between heartbeat frame to request of the broker. A value of 0 disables heartbeats. Note rabbitmq-c only has partial support for hearts, as of v0.4.0 heartbeats are only serviced during BasicPublish(), and SimpleWaitFrame() properties a table of properties to send the broker.
sasl_method the SASL method to authenticate with the properties.

Returns error code indicating success or failure via RabbitMQRPCReplyMBS object.

- Reply.Type = kResponseNormal. Login completed successfully

- Reply.Type = kResponseLibraryException. In most cases errors from the broker when logging in will be represented by the broker closing the socket. In this case r.libraryError will be set to kStatusConnectionClosed. This error can represent a number of error conditions including: invalid vhost, authentication failure.

- Reply.Type = kResponseServerException. The broker returned an exception:

- If Reply.MethodId = kChannelCloseMethod a channel exception occurred, check decoded properties to see details of the exception. The client should SendMethod() a kChannelCloseOkMethod. The channel must be re-opened before it can be used again. Any resources associated with the channel (auto-delete exchanges, auto-delete queues, consumers) are invalid and must be recreated before attempting to use them again.

- If Reply.MethodId = kConnectionCloseMethod a connection exception occurred, check decoded properties to see details of the exception. The client SendMethod() a kConnectionCloseOkMethod and disconnect from the broker.

3.3.28 LoginPlain(vhost as String, channelMax as Integer, frameMax as Integer, heartbeat as Integer, Username as String, Password as String, properties as Dictionary = nil) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Login with plain SASL.
Notes: Login to the broker passing an optional properties table.

vhost: the virtual host to connect to on the broker. The default on most brokers is "/".
channelMax: the limit for the number of channels for the connection.
0 means no limit, and is a good default (kDefaultMaxChannels) Note that the maximum number of channels the protocol supports is 65535 (2^16, with the 0-channel reserved). The server can set a lower channelMax and then the client will use the lowest of the two.
frameMax: the maximum size of an AMQP frame ont he wire to request of the broker for this connection. 4096 is the minimum size, 2^31-1 is the maximum, a good default is 131072 (128KB), or kDefaultFrameSize.
heartbeat: the number of seconds between heartbeat frame to request of the broker. A value of 0 disables heartbeats. Note rabbitmq-c only has partial support for hearts, as of v0.4.0 heartbeats are only serviced during BasicPublish(), and SimpleWaitFrame() properties a table of properties to send the broker.
sasl_method the SASL method to authenticate with the properties.

Returns error code indicating success or failure via RabbitMQRPCReplyMBS object.

- Reply.Type = kResponseNormal. Login completed successfully
- Reply.Type = kResponseLibraryException. In most cases errors from the broker when logging in will be represented by the broker closing the socket. In this case r.libraryError will be set to kStatusConnectionClosed. This error can represent a number of error conditions including: invalid vhost, authentication failure.
- Reply.Type = kResponseServerException. The broker returned an exception:
- If Reply.MethodId = kChannelCloseMethod a channel exception occurred, check decoded properties to see details of the exception. The client should SendMethod() a kChannelCloseOkMethod. The channel must be re-opened before it can be used again. Any resources associated with the channel (auto-delete exchanges, auto-delete queues, consumers) are invalid and must be recreated before attempting to use them again.
- If Reply.MethodId = kConnectionCloseMethod a connection exception occurred, check decoded properties to see details of the exception. The client SendMethod() a kConnectionCloseOkMethod and disconnect from the broker.

3.3.29 MaybeReleaseBuffers

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Release connection owned memory.
Notes: Release memory owned by the connection object related to any channel, allowing reuse by the library. Use of any memory returned by the library before this function is called with result in undefined behavior.

Internally rabbitmq-c tries to reuse memory when possible. As a result its possible calling this function may not have a noticeable effect on memory usage.

3.3.30 MaybeReleaseBuffersOnChannel(channel as Integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Release connection owned memory related to a channel.
Notes: Release memory owned by the amqp_connection_state_t object related to the specified channel, allowing reuse by the library. Use of any memory returned the library for a specific channel will result in undefined behavior.

Internally rabbitmq-c tries to reuse memory when possible. As a result its possible calling this function may not have a noticeable effect on memory usage.

3.3.31 MethodHasContent(constantNumber as Integer) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks whether such a method has content.
Notes: Returns true if content is expected.

3.3.32 MethodName(constantNumber as Integer) as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get method name string from method number.
Notes: Returns method name string.

3.3.33 NewTCPSocket(UseSSL as boolean = false) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates new socket.
Notes: Optional uses OpenSSL to create a SSL socket if UseSSL is passed with value true.

Returns true on success.

3.3.34 OpenSocket(Host as String, Port as Integer, TimeOut as Double = 10) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Open a socket connection.
Notes: This function opens a socket connection returned created with NewTCPSocket(). This function should be called after setting socket options.
Returns error code.

host: Connect to this host.
port: Connect on this remote port.
timeout: Max allowed time to spent on opening.

3.3.35 QueueBind(channel as Integer, queue as String, exchange as String, routingKey as String, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Binds an exchange.

3.3.36 QueueDeclare(channel as Integer, queue as String, passive as Boolean, durable as Boolean, exclusive as Boolean, autoDelete as Boolean, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Declare a queue.

3.3.37 QueueDelete(channel as Integer, queue as String, ifUnused as Boolean, ifEmpty as Boolean) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Delete a queue.

3.3.38 QueuePurge(channel as Integer, queue as String) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Purge a queue.

3.3.39 QueueUnbind(channel as Integer, queue as String, exchange as String, routingKey as String, arguments as Dictionary) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Unbind a queue.

3.3.40 ReadMessage(channel as Integer, byref message as RabbitMQMessageMBS, flags as Integer = 0) as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reads the next message on a channel
Notes: Reads a complete message (header + body) on a specified channel. This function is intended to be used with amqp_basic_get() or when an AMQP_BASIC_DELIVERY_METHOD method is received.

channel: the channel on which to read the message from
message: The parameter receiving the message.
flags: currently unused.

Returns RabbitMQRPCReplyMBS, which may contain error information.
Type of the object is RabbitMQRPCReplyMBS.ResponseTypeNormal on success.

3.3.41 SendHeader as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Send initial AMQP header to the broker.
Notes: This is a low level function intended for those who want to interact with the broker at a very low level. Use of this function without understanding what it does will result in AMQP protocol errors.

This function sends the AMQP protocol header to the broker.

Returns error code, kStatusOK on success, a negative value on failure. Possible error codes:

- kStatusConnectionClosed: the connection to the broker was closed.
- kStatusSocketError: a socket error occurred. It is likely the underlying socket has been closed. errno or WSAGetLastError() may provide further information.
- kStatusSSLError: a SSL error occurred. The connection to the broker was closed.

3.3.42 SetCACert(File as FolderItem) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets CACert file with root certificates.
Notes: Returns true on success and false on failure.
See also:

- 3.3.43 SetCACert(Path as String) as Boolean 33

3.3.43 SetCACert(Path as String) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets CACert file with root certificates.
Notes: Returns true on success and false on failure.
See also:

- 3.3.42 SetCACert(File as FolderItem) as Boolean 33

3.3.44 SetKey(CertFile as FolderItem, KeyFile as FolderItem) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets SSL key with certificate file path and key file path.
Notes: Returns true on success and false on failure.
See also:

- 3.3.45 SetKey(CertPath as String, KeyPath as String) as Boolean 33

3.3.45 SetKey(CertPath as String, KeyPath as String) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets SSL key with certificate file path and key file path.
Notes: Returns true on success and false on failure.
See also:

- 3.3.44 SetKey(CertFile as FolderItem, KeyFile as FolderItem) as Boolean 33

3.3.46 SetKeyData(CertFile as FolderItem, KeyData as String) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets SSL key with certificate file and key data as PEM.
Notes: Returns true on success and false on failure.
See also:

- 3.3.47 SetKeyData(CertPath as String, KeyData as String) as Boolean 34

3.3.47 SetKeyData(CertPath as String, KeyData as String) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets SSL key with certificate file and key data as PEM.
Notes: Returns true on success and false on failure.
See also:

- 3.3.46 SetKeyData(CertFile as FolderItem, KeyData as String) as Boolean 34

3.3.48 SimpleWaitFrame(byref Frame as RabbitMQFrameMBS, timeoutValue as double = 30) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Read a single frame with a timeout.
Notes: Waits for the next frame to be read from the broker, up to a timespan specified by tv. The function will return kStatusTimeout if the timeout is reached. The tv value is not modified by the function.

The library may buffer frames. When an connection object has frames buffered calling SimpleWaitFrame_noblock() will return an frame without entering a blocking read(). You can test to see if an connection object has frames buffered by calling the FramesEnqueued() function.

The library has a socket read buffer. When there is data in an connection read buffer, SimpleWaitFrame() may return an frame without entering a blocking read(). You can test to see if an connection object has data in its read buffer by calling the DataInBuffer() function.

This function does not return heartbeat frames. When enabled, heartbeating is handed internally internally by the library.

Frame: receives decoded frame the frame
timeoutValue: the maximum time to wait for a frame to be read in seconds. Setting to 0 will do a non-blocking read.

Returns kStatusOK on success. An kStatus value is returned otherwise. Possible errors include:

- kStatusTimeout the timeout was reached while waiting for a frame from the broker.
- kStatusInvalidParameter the tv parameter contains an invalid value.
- kStatusNoMemory failure in allocating memory. The library is likely in an indeterminate state making recovery unlikely. Client should note the error and terminate the application
- kStatusBadAMQPData bad AMQP data was received. The connection should be shutdown immediately
- kStatusUnknownMethod: an unknown method was received from the broker. This is likely a protocol error and the connection should be shutdown immediately
- kStatusUnknownClass: a properties frame with an unknown class was received from the broker. This is likely a protocol error and the connection should be shutdown immediately
- kStatusHeartbeatTimeout: timed out while waiting for heartbeat from the broker. The connection has been closed.
- kStatusTimerFailure: system timer indicated failure.
- kStatusSocketError: a socket error occurred. The connection has been closed
- kStatusSSLError: a SSL socket error occurred. The connection has been closed.

3.3.49 TXCommit(channel as Integer) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Commit.

3.3.50 TXRollback(channel as Integer) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Rollback.

3.3.51 TXSelect(channel as Integer) as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Select.

3.3.52 Properties

3.3.53 ClientProperties as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the client properties table
Notes: Get the properties that were passed to the broker on connection.
(Read only property)

3.3.54 DataInBuffer as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Check to see if there is data left in the receive buffer.
Notes: Can be used to see if there is data still in the buffer, if so calling SimpleWaitFrame will not immediately enter a blocking read.

Returns true if there is data in the receive buffer, false otherwise.
(Read only property)

3.3.55 FramesEnqueued as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks to see if there are any incoming frames ready to be read.
Notes: Checks to see if there are any frame objects buffered by the connection object. Having one or more frames buffered means that SimpleWaitFrame() will return a frame without potentially blocking on a read() call.

Returns true if there are frames enqueued, false otherwise.
(Read only property)

3.3.56 Handle as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The internal object reference.
Notes: (Read and Write property)

3.3.57 HandshakeTimeout as Double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get or set the login handshake timeout.
Notes: Login performs the login handshake with the broker. This function sets the timeout associated with completing this operation from the client side.

The timeout must be set before Login is called to change from the default timeout.

Note that the RabbitMQ broker has a configurable timeout for completing the login handshake, the default is 10 seconds. rabbitmq-c has a default of 12 seconds.
(Read and Write property)

3.3.58 Heartbeat as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the number of seconds between heartbeats of the connection.
Notes: The number of seconds between heartbeats is set when connection negotiation takes place in Login.

Returns the number of seconds between heartbeats.
(Read only property)

3.3.59 MaximumChannel as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the maximum number of channels the connection can handle.
Notes: The maximum number of channels is set when connection negotiation takes place in Login.

Returns the maximum number of channels. 0 if there is no limit.
(Read only property)

3.3.60 MaximumFrameSize as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the maximum size of an frame the connection can handle.
Notes: The maximum size of an frame is set when connection negotiation takes place in Login.

Return the maximum size in bytes of an frame.
(Read only property)

3.3.61 RPCReply as RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the last global RPCReply
Notes: The API methods corresponding to most synchronous AMQP methods return a pointer to the decoded method result. Upon error, they return nil, and we need some way of discovering what, if anything, went wrong. RPCReply() returns the most recent RabbitMQRPCReplyMBS instance corresponding to such an API operation for the given connection.

Only use it for operations that do not themselves return RabbitMQRPCReplyMBS; operations that do return RabbitMQRPCReplyMBS generally do NOT update this per-connection-global RabbitMQRPCReplyMBS instance.

Returns the most recent RabbitMQRPCReplyMBS object:

- Reply.Type = kResponseNormal. Login completed successfully
- Reply.Type = kResponseLibraryException. In most cases errors from the broker when logging in will be represented by the broker closing the socket. In this case r.libraryError will be set to kStatusConnectionClosed. This error can represent a number of error conditions including: invalid vhost, authentication failure.
- Reply.Type = kResponseServerException. The broker returned an exception:
- If Reply.MethodId = kChannelCloseMethod a channel exception occurred, check decoded properties to see details of the exception. The client should SendMethod() a kChannelCloseOkMethod. The channel must be re-opened before it can be used again. Any resources associated with the channel (auto-delete exchanges, auto-delete queues, consumers) are invalid and must be recreated before attempting to use them again.
- If Reply.MethodId = kConnectionCloseMethod a connection exception occurred, check decoded properties to see details of the exception. The client SendMethod() a kConnectionCloseOkMethod and disconnect from the broker.

(Read only property)

3.3.62 RPCTimeout as Double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The RPC timeout.
Notes: Get or set the RPC timeout.

Sets the timeout for any RPC-style AMQP command (e.g., QueueDeclare).
This timeout may be changed at any time by calling this function with a new timeout. The timeout applies individually to each RPC that is made.

The default value is 0, or an infinite timeout.

When an RPC times out, the function will return an error kStatusTimeout, and the connection will be closed.

RPC-timeouts are an advanced feature intended to be used to detect dead connections quickly when the rabbitmq-c implementation of heartbeats does not work. Do not use RPC timeouts unless you understand the implications of doing so.
(Read and Write property)

3.3.63 ServerProperties as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the broker properties table.
Notes: Returns a dictionary containing the properties advertised by the broker on connection.
(Read only property)

3.3.64 SocketFileDescriptor as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Get the socket descriptor in use by a socket object.
Notes: Retrieve the underlying socket descriptor. This function can be used to perform low-level socket operations that aren't supported by the socket interface. Use with caution!
(Read only property)

3.3.65 SSLCACertPath as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The CA certificate file path.
Notes: cacert Path to the CA cert file in PEM format.
(Read only property)

3.3.66 SSLCertPath as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The certificate file path.
Notes: Cert file in PEM format.
(Read only property)

3.3.67 SSLKeyData as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The client key.
Notes: A buffer containing client key in PEM format.
(Read only property)

3.3.68 SSLKeyPath as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The file path to the SSL key.
Notes: (Read only property)

3.3.69 SSLMaxVersion as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Maximum allowed SSL version.
Notes: Defaults to latest TLS version.
(Read and Write property)

3.3.70 SSLMinVersion as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Minimum allowed SSL version.
Notes: Defaults to TLS v1.0.
(Read and Write property)

3.3.71 SSLVerifyHostname as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Enable or disable hostname verification.
Notes: Hostname verification checks the broker cert for a CN or SAN that matches the hostname that OpenSocket() is presented.
Peer verification is controlled by SSLVerifyPeer property.
(Read and Write property)

3.3.72 SSLVerifyPeer as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Enable or disable peer verification.
Notes: Peer verification validates the certificate chain that is sent by the broker.
Hostname validation is controlled by SSLVerifyHostname.
(Read and Write property)

3.3.73 Version as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries version number of the library.
Notes: Returns the rabbitmq-c version as a string.
(Read only property)

3.3.74 VersionNumber as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries version number of the library.
Notes: Returns the rabbitmq-c version as a packed integer.

first byte is major, second byte minor, third byte patch level and last release counter.
Value = ((((Major * 256) + Minor) * 256 + Patch) * 256 + Release)
(Read only property)

3.3.75 Constants

Constants

Constant	Value	Description
kDefaultFrameSize	131072	Default frame size (128Kb)
kDefaultHeartbeat	0	Default heartbeat interval (0, heartbeat disabled)
kDefaultVHost	"/"	Default RabbitMQ vhost: "/".
kMaxChannels	2047	Default maximum number of channels (2047, RabbitMQ default limit of 2048, minus 1 for channel 0). RabbitMQ set a default limit of 2048 channels per connection in v3.7.5 to prevent broken clients from leaking too many channels.
kProtocolPort	5672	Default AMQP Port.
kProtocolVersionMajor	0	AMQP protocol version major
kProtocolVersionMinor	9	AMQP protocol version minor.
kProtocolVersionRevision	1	AMQP protocol version revision.

Status constants.

Constant	Value	Description
kStatusBadAmqpData	-2	Incorrect or corrupt data was received from the broker. This is a protocol
kStatusBadUrl	-8	malformed AMQP URL
kStatusBrokerUnsupportedSaslMethod	-19	Broker does not support the requested SASL mechanism.
kStatusConnectionClosed	-7	The connection to the broker has been closed.
kStatusHeartbeatTimeout	-15	Timed out waiting for heartbeat.
kStatusHostnameResolutionFailed	-5	Unable to resolve the hostname.
kStatusIncompatibleAmqpVersion	-6	The broker advertised an incompaible AMQP version.
kStatusInvalidParameter	-10	An invalid parameter was passed into the function.
kStatusNoMemory	-1	Memory allocation failed.
kStatusOk	0	Operation successful.
kStatusSocketClosed	-17	Underlying socket is closed.
kStatusSocketError	-9	A socket error occurred.
kStatusSocketInuse	-18	Underlying socket is already open.
kStatusSslConnectionFailed	-515	SSL handshake failed.
kStatusSslError	-512	A generic SSL error occurred.
kStatusSslHostnameVerifyFailed	-513	SSL validation of hostname against peer certificate failed.
kStatusSslPeerVerifyFailed	-514	SSL validation of peer certificate failed.
kStatusTableTooBig	-11	The dictionary object cannot be serialized because the output buffer small.
kStatusTcpError	-256	A generic TCP error occurred.
kStatusTcpSocketlibInitError	-257	An error occurred trying to initialize the socket library
kStatusTimeout	-13	Operation timed out.
kStatusTimerFailure	-14	The underlying system timer facility failed.
kStatusUnexpectedState	-16	Unexpected protocol state
kStatusUnknownClass	-3	An unknown AMQP class was received. This is a protocol error.
kStatusUnknownMethod	-4	An unknown AMQP method was received. This is a protocol error.
kStatusUnsupported	-20	Parameter is unsupported in this version.
kStatusWrongMethod	-12	The wrong method was received.

TLS Version

Constant	Value	Description
kTLSv10	1	TLS v1.0
kTLSv11	2	TLS v1.1
kTLSv12	3	TLS v1.2
kTLSvLatest	65535	Latest TLS version available.

3.4 class RabbitMQEnvelopeMBS

3.4.1 class RabbitMQEnvelopeMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Envelope object.
Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.4.2 Methods

3.4.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The private constructor.

3.4.4 Destructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.

3.4.5 Properties

3.4.6 Channel as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Channel message was delivered on.
Notes: (Read and Write property)

3.4.7 ConsumerTag as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The consumer tag the message was delivered to .
Notes: (Read and Write property)

3.4.8 DeliveryTag as UInt64

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The messages delivery tag.
Notes: (Read and Write property)

3.4.9 Exchange as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The exchange this message was published to.
Notes: (Read and Write property)

3.4.10 Message as RabbitMQMessageMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The message.
Notes: (Read and Write property)

3.4.11 Redelivered as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Flag indicating whether this message is being redelivered.
Notes: (Read and Write property)

3.4.12 RoutingKey as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The routing key this message was published with.
Notes: (Read and Write property)

3.5 class RabbitMQFrameMBS

3.5.1 class RabbitMQFrameMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: An AMQP frame.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.5.2 Methods

3.5.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The constructor.

3.5.4 Destructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.

3.5.5 Properties

3.5.6 BodyFragment as MemoryBlock

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The body fragment.
Notes: Only set if FrameType is kFrameBody.
(Read and Write property)

3.5.7 BodyFragmentSize as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The size of the body fragment in bytes.
Notes: Only set if FrameType is kFrameBody.
(Read and Write property)

3.5.8 BodySize as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Size of the body in bytes.
Notes: Only set if FrameType is kFrameHeader.
(Read and Write property)

3.5.9 Channel as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The channel the frame was received on.
Notes: (Read and Write property)

3.5.10 ClassID as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for the properties.
Notes: Only set if FrameType is kFrameHeader.
(Read and Write property)

3.5.11 FrameType as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The frame type.
Notes: See kFrameMethod, kFrameHeader and kFrameBody.
(Read and Write property)

3.5.12 MethodDecoded as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The decoded method content.
Notes: MBS Plugin has code to translate over 60 types of method structures into Xojo dictionary.
Only set if FrameType is kFrameMethod.
(Read and Write property)

3.5.13 MethodID as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The method ID.
Notes: Only set if FrameType is kFrameMethod.
(Read and Write property)

3.5.14 PropertiesDecoded as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The decoded properties.
Notes: MBS Plugin has code to translate over several types of properties structures into Xojo dictionary.
Only set if FrameType is kFrameHeader.
(Read and Write property)

3.5.15 Raw as MemoryBlock

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Amqp-encoded properties structure.
Notes: Only set if FrameType is kFrameHeader.
(Read and Write property)

3.5.16 RawSize as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The size of the Raw data.
Notes: Only set if FrameType is kFrameHeader.
(Read and Write property)

3.5.17 Constants

Constants

Constant	Value	Description
kFrameMinSize	4096	Minimum frame size.

Method IDs

Constant	Value	Description
kAccessRequestMethod	&h001E000A	
kAccessRequestOkMethod	&h001E000B	
kBasicAckMethod	&h003C0050	
kBasicCancelMethod	&h003C001E	
kBasicCancelOkMethod	&h003C001F	
kBasicConsumeMethod	&h003C0014	
kBasicConsumeOkMethod	&h003C0015	
kBasicDeliverMethod	&h003C003C	
kBasicGetEmptyMethod	&h003C0048	
kBasicGetMethod	&h003C0046	
kBasicGetOkMethod	&h003C0047	
kBasicNackMethod	&h003C0078	
kBasicPublishMethod	&h003C0028	
kBasicQosMethod	&h003C000A	
kBasicQosOkMethod	&h003C000B	
kBasicRecoverAsyncMethod	&h003C0064	
kBasicRecoverMethod	&h003C006E	
kBasicRecoverOkMethod	&h003C006F	
kBasicRejectMethod	&h003C005A	
kBasicReturnMethod	&h003C0032	
kChannelCloseMethod	&h00140028	
kChannelCloseOkMethod	&h00140029	
kChannelFlowMethod	&h00140014	
kChannelFlowOkMethod	&h00140015	
kChannelOpenMethod	&h0014000A	
kChannelOpenOkMethod	&h0014000B	
kConfirmSelectMethod	&h0055000A	
kConfirmSelectOkMethod	&h0055000B	
kConnectionBlockedMethod	&h000A003C	
kConnectionCloseMethod	&h000A0032	
kConnectionCloseOkMethod	&h000A0033	
kConnectionOpenMethod	&h000A0028	
kConnectionOpenOkMethod	&h000A0029	
kConnectionSecureMethod	&h000A0014	
kConnectionSecureOkMethod	&h000A0015	
kConnectionStartMethod	&h000A000A	
kConnectionStartOkMethod	&h000A000B	
kConnectionTuneMethod	&h000A001E	
kConnectionTuneOkMethod	&h000A001F	
kConnectionUnblockedMethod	&h000A003D	
kExchangeBindMethod	&h0028001E	
kExchangeBindOkMethod	&h0028001F	
kExchangeDeclareMethod	&h0028000A	
kExchangeDeclareOkMethod	&h0028000B	
kExchangeDeleteMethod	&h00280014	
kExchangeDeleteOkMethod	&h00280015	
kExchangeUnbindMethod	&h00280028	
kExchangeUnbindOkMethod	&h00280033	
kQueueBindMethod	&h00320014	
kQueueBindOkMethod	&h00320015	
kQueueDeclareMethod	&h0032000A	
kQueueDeclareOkMethod	&h0032000B	
kQueueDeleteMethod	&h00320028	
kQueueDeleteOkMethod	&h00320029	
kQueuePurgeMethod	&h0032001E	
kQueuePurgeOkMethod	&h0032001F	
kQueueUnbindMethod	&h00320032	
kQueueUnbindOkMethod	&h00320033	

Properties classes

Constant	Value	Description
kClassAccess	&h001E	access class id
kClassBasic	&h003C	basic class id
kClassChannel	&h0014	channel class id
kClassConection	&h000A	connection class id
kClassConfirm	&h0055	confirm class id
kClassExchange	&h0028	exchange class id
kClassQueue	&h0032	queue class id
kClassTX	&h005A	tx class id

Frame Types

Constant	Value	Description
kFrameBody	3	The BodyFragment and BodyFragmentSize properties are set.
kFrameEnd	206	Frame end type.
kFrameHeader	2	ClassID, BodySize, PropertiesDecoded, Raw and RawSize properties are set.
kFrameHeartbeat	8	Heartbeat frame type.
kFrameMethod	1	The MethodDecoded and MethodID properties are set.

3.6 class RabbitMQMessageMBS

3.6.1 class RabbitMQMessageMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for a message.
Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.6.2 Methods

3.6.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The private construtor.

3.6.4 Properties

3.6.5 Body as MemoryBlock

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The message body.
Notes: (Read only property)

3.6.6 Properties as RabbitMQBasicPropertiesMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The message properties.
Notes: (Read only property)

3.6.7 Constants

Response Types

Constant	Value	Description
ResponseTypeLibraryException	2	Library error, an error occurred in the library, examine the library error.
ResponseTypeNone	0	The library got an EOF from the socket.
ResponseTypeNormal	1	Response normal, the RPC completed successfully.
ResponseTypeServerException	3	Server exception, the broker returned an error, check reply.

3.7 class RabbitMQRPCReplyMBS

3.7.1 class RabbitMQRPCReplyMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Reply from a RPC method on the broker.
Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
Blog Entries

- News from the MBS Xojo Plugins Version 21.2
- Bringing RabbitMQ to Xojo

3.7.2 Methods

3.7.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The private constructor.

3.7.4 Destructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.

3.7.5 Properties

3.7.6 LibraryError as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The library error code.
Notes: in case of ResponseTypeLibraryException this field will be set to an error code.
An error string can be retrieved using ErrorString().
(Read only property)

3.7.7 MethodDecoded as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The decoded data for the method.
Notes: MBS Plugin has code to translate over 60 types of method structures into Xojo dictionary.
(Read only property)

3.7.8 MethodId as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The method ID.
Notes: (Read only property)

3.7.9 Type as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The response type.
Notes: See ResponseTypeLibraryException, ResponseTypeServerException, ResponseTypeNone and ResponseTypeNormal.
(Read only property)

3.7.10 Constants

Error Codes

Constant	Value	Description
kAccessRefused	403	
kChannelError	504	
kCommandInvalid	503	
kConnectionForced	320	
kContentTooLarge	311	
kFrameError	501	
kInternalError	541	
kInvalidPath	402	
kNoConsumers	313	
kNoRoute	312	
kNotAllowed	530	
kNotFound	404	
kNotImplemented	540	
kPreconditionFailed	406	
kReplySuccess	200	Success
kResourceError	506	
kResourceLocked	405	
kSyntaxError	502	
kUnexpectedFrame	505	

Response Types

Constant	Value	Description
ResponseTypeLibraryException	2	The library encountered an error, check the LibraryError field.
ResponseTypeNone	0	Invalid state.
ResponseTypeNormal	1	The RPC completed successfully.
ResponseTypeServerException	3	The broker returned an exception, check the reply field.