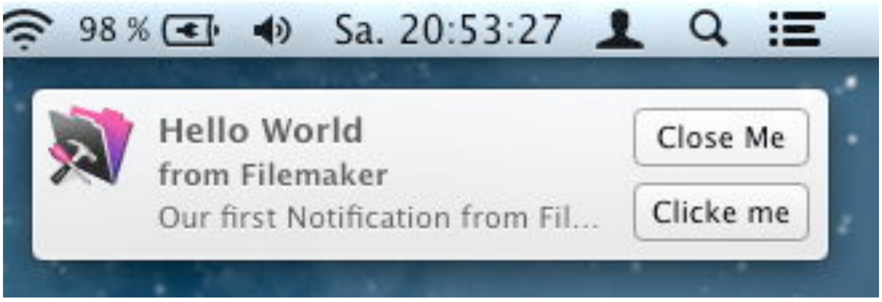
Mac OS X 10.8 User Notifications with Filemaker and Real Studio


Hello World
from Filemaker
Our first Notification from Fil...

With Mountain Lion you can now post user notifications to the user. Depending on the settings defined in system preferences, your notification shows an alert, a banner or nothing. You can play a sound and user can see the notifications. You can schedule notifications to fire while your app is in background or not running. When clicked on the notification, the system can launch your application and inform you which notification was clicked.

So let's look into it. Our plugins have functions for Filemaker (in new 2.8 plugins) and classes for Real Studio (in new 12.3 plugins).

First we need to create a new notification object. In Filemaker, we set a variable with the new reference number:

```
Set Variable [$ref; MBS("UserNotification.Create")]
```

And in Real Studio we create a new user notification object:

```
dim u as new NSUserNotificationMBS
```

Next we can define title, subtitle and the informative text. For that we use three more set variable steps in Filemaker. We pass the reference number and also put return values in result variables, so you can see the variables in debugger and notice any error message from the plugin.

```
Set Variable [$result; MBS("UserNotification.SetTitle"; $ref; "Hello World")]
Set Variable [$result; MBS("UserNotification.SetSubTitle"; $ref; "from Filemaker")]
Set Variable [$result; MBS("UserNotification.SetInformativeText"; $ref; "Our first Notification from Filemaker. ")]
```

In Real Studio we simply assign the properties with the texts:

```
u.Title = "Hello World"
u.subtitle = "from Real Studio."
u.informativeText = "Our first Notification from Real Studio."
```

You can customize the notification with an action button, but that button is only visible if user allows you to show alerts. In that case it is interesting to register a callback in Filemaker or use the NSUserNotificationCenterDelegateMBS class in Real Studio to get notified if button is pressed or notification content has been clicked.

For the notification you can also define a sound to be played and you can store user information data. That's text like a record ID which you can put into the notification and retrieve later when you get an event that button is clicked or notification has been delivered. Also you can query scheduled and delivered notifications and check the user info there to update your own database for what the user saw.

You can show a notification with the deliver function, but we really want to schedule the notification to show later. So for this example we query current date and time and simply add ten seconds. In Filemaker, you write this like this:

```
MBS("UserNotification.SetDeliveryDate"; $ref; Get
( CurrentHostTimeStamp ) + 10)
MBS("UserNotification.scheduleNotification"; $ref)
```

In Real Studio we use the date class to get current date, add ten seconds and set delivery date. To deliver or schedule we need an object of the NSUserNotificationCenterMBS class, so we create an instance and call the scheduleNotification method:

```
dim d as new date
d.Second = d.Second + 10
u.deliveryDate = d

dim c as new NSUserNotificationCenterMBS
c.scheduleNotification u
```

Finally, to free memory, you can call UserNotification.Release in Filemaker. In Real Studio simply let the objects go out of scope.

Also important: Your Filemaker solution or Real Studio applications need to have an unique application identifier. So for Real Studio, please make sure app.ApplicationIdentifier is set. On Filemaker, the app itself has an identifier, but runtimes may need to have it customized, so not all runtimes have same identifier. You find the bundle identifier in the info.plist inside the application package.

I hope you enjoy notifications and if you have questions, please do not hesitate to ask us.