



MMScript

MMScript_AreScriptEventsEnabled

MMScript_CancelScriptEvent

MMScript_Configure

MMScript_CreateDateTimeScriptEvent

MMScript_CreateIdleScriptEvent

MMScript_CreateImmediateScriptEvent

MMScript_CreateIntervalScriptEvent

MMScript_CreateXMLScriptEvent

MMScript_DisableScriptEvents

MMScript_EnableScriptEvents

MMScript_GetScriptEventName

MMScript_GetScriptEventValue

MMScript_IsUserIdle

MMScript_ListScriptEvents

MMScript_Register

MMScript_ShellExecute

MMScript_UpdateScriptEventValue

MMScript_Version

MMScript_VersionAutoUpdate

Reference_XML

MMScript_AreScriptEventsEnabled

Description

This function will return 1 if the plug-in is currently processing Script Events or 0 if not. You can use MMScript_DisableScriptEvents and MMScript_EnableScriptEvents to tell the plug-in to process Script Events or not.

Return Type

Number (1=True, 0=False)

Format

`MMScript_AreScriptEventsEnabled`

Related Items

MMScript_DisableScriptEvents, MMScript_EnableScriptEvents

MMScript_CancelScriptEvent

Description

Use this function to Cancel an existing Script Event.

Return Type

Text

Format

MMScript_CancelScriptEvent (**EventName**)

Required Parameters

EventName

The Name of the Event you want to Cancel.

Related Items

MMScript_CreateDateTimeScriptEvent, MMScript_CreateIdleScriptEvent,
MMScript_CreateImmediateScriptEvent, MMScript_CreateIntervalScriptEvent,
MMScript_CreateXMLScriptEvent, MMScript_ListScriptEvents

MMScript_Configure

Description

Calling this function with no parameters will open the MMScript Configuration Dialog. You can optionally open to a specific tab by specifying the name of the tab as the "Option" parameter. This function also allows you to get or set any preference found in the Configuration Dialog.

Valid PrefNames:

"AddHelp" - If True, Function specific Help will be added to Calculations when inserting the Plug-in's Functions.

"ScriptsEnabled" - If False, the plug-in will not call scripts in your database.

"UseOldFunctions" - If True, the old-style External("Scrp-XXX") functions will be available.

"UseNewFunctions" - If True, the new-style MMScript_XXX functions will be available.

Return Type

Varies

Format

MMScript_Configure (**Option** ; **PrefName** ; **PrefValue**)

Optional Parameters

Option

Specify the name of a Tab in the Configuration Dialog to show it opened to that tab.

Specify "Get" with the PrefName parameter to get a preference value. (If not found, and PrefValue is defined, PrefValue will be returned.)

Specify "Set" with the PrefName and PrefValue parameters to set a preference value.

PrefName

The Name of the Preference to Get or Set. (See the Function Description for a list of valid PrefNames.)

PrefValue

The Value of the Preference to Set. (See the Function Description for some possible values.)

Examples

Example 1

Code:

```
MMScript_Configure
```

Result:

Opens the MMScript Configuration Dialog. (Because the "Option" parameter is not used, the dialog will open to the "Basics" tab.)

Example 2

Code:

```
MMScript_Configure( "About" )
```

Result:

Opens the MMScript Configuration Dialog to the "About" tab.

Example 3

Code:

```
MMScript_Configure( "Get" ; "AddHelp" )
```

Result:

Returns the value of the 'Add Help Comments to External Functions' setting from the Configuration Dialog.

Example 4

Code:

```
MMScript_Configure( "Set" ; "ScriptsEnabled" ; "True" )
```

Result:

Sets the 'Enable Script Events' setting in the Configuration Dialog to true.

MMScript_CreateDateTimeScriptEvent

Description

Use this function to create a Script Event that occurs at a specific date and/or time in the future.

Return Type

Text

Format

MMScript_CreateDateTimeScriptEvent (**EventName** ; **Script** ; **DateTime** ; **DB** ; **EventValue** ; **Persistent** ; **Priority** ; **CurrentScript**)

Required Parameters

EventName

The Name of the Event you are Creating.

Script

The Script to call.

DateTime

A Date, Time, or Timestamp of when you want the Event to occur. If you are passing a text string to this parameter, you may need to convert it using either GetAsDate(), GetAsTime(), or GetAsTimestamp(). See the examples section to see what this looks like.

Optional Parameters

DB

The Database that contains the Script.

EventValue

Any user defined Value you can retrieve when the Event occurs.

Persistent

True or False if you want this Event to be saved between FileMaker sessions.

Priority

Defines how interruptive the Event is. (Specify "Low", "Normal", "High", or "Highest".)

Low - The Event occurs if the user has not done anything for 30 seconds or more.

Normal - The Event may occur even if the user has done something within the last 30 seconds, but no script is running.

High - The Event may occur if the user is running a script that is currently paused.

Highest - The Event attempts to run even if there is already a script running.

CurrentScript

Defines what to do with the currently running script (if any). (Specify "Halt", "Exit", "Resume", or "Pause".)

Related Items

MMScript_CancelScriptEvent, MMScript_CreateIdleScriptEvent,

MMScript_CreateImmediateScriptEvent, MMScript_CreateIntervalScriptEvent,
MMScript_CreateXMLScriptEvent, MMScript_ListScriptEvents

Example

Code:

```
MMScript_CreateDateTimeScriptEvent ( "My Event"; "My Script" ; GetAsTimestamp("9/8/2011 3:
```

Result:

Creates an event named "My Event" to call the "My Script" script at 9/8/2011 3:49:35 PM. FileMaker's GetAsTimestamp() function is used to turn the text string of "9/8/2011 3:49:35 PM" into a timestamp. GetAsDate() and GetAsTime() can also be used here if you only want to specify a Date or Time.

MMScript_CreateIdleScriptEvent

Description

Use this function to create a Script Event that occurs after the user has been idle for a specific amount of time. A Repeating Idle Script Event will only occur once per idle state. In other words, say you have a Script Event set to occur when the user is idle for 30 seconds, and a user went idle at 10:01:00. Your Idle Script would be called at 10:01:30 but would not be called at 10:02:00 or 10:02:30, etc. Only after the user became active again and then idle again for 30 seconds would your Idle Script be called again.

Return Type

Text

Format

MMScript_CreateIdleScriptEvent (**EventName** ; **Script** ; **IdleTime** ; **DB** ; **EventValue** ; **Repeat** ; **Persistent**)

Required Parameters

EventName

The Name of the Event you are Creating.

Script

The Script to call.

IdleTime

The amount of time the user must be idle before this Event occurs (Format: "1h 2d 3m 45s").

Optional Parameters

DB

The Database that contains the Script.

EventValue

Any user defined Value you can retrieve when the Event occurs.

Repeat

True or False if you want the Event to Repeat until you cancel it.

Persistent

True or False if you want this Event to be saved between FileMaker sessions.

Related Items

MMScript_CancelScriptEvent, MMScript_CreateDateTimeScriptEvent,
MMScript_CreateImmediateScriptEvent, MMScript_CreateIntervalScriptEvent,
MMScript_CreateXMLScriptEvent, MMScript_IsUserIdle, MMScript_ListScriptEvents

MMScript_CreateImmediateScriptEvent

Description

Use this function to call a script immediately. This is useful for processing fields when a user leaves them or for calling a dynamically named script.

Return Type

Text

Format

MMScript_CreateImmediateScriptEvent (**Script** ; **DB** ; **EventName** ; **EventValue** ; **Priority** ; **CurrentScript**)

Required Parameters

Script

The Script to call.

Optional Parameters

DB

The Database that contains the Script.

EventName

The Name of the Event you are creating.

EventValue

Any user defined Value you can retrieve when the Event occurs.

Priority

Defines how interruptive the Event is. (Specify "Low", "Normal", "High", or "Highest".)

Low - The Event occurs if the user has not done anything for 30 seconds or more.

Normal - The Event may occur even if the user has done something within the last 30 seconds, but no script is running.

High - The Event may occur if the user is running a script that is currently paused.

Highest - The Event attempts to run even if there is already a script running.

CurrentScript

Defines what to do with the currently running script (if any). (Specify "Halt", "Exit", "Resume", or "Pause".)

Related Items

MMScript_CancelScriptEvent, MMScript_CreateDateTimeScriptEvent,
MMScript_CreateIdleScriptEvent, MMScript_CreateIntervalScriptEvent,
MMScript_CreateXMLScriptEvent, MMScript_ListScriptEvents

MMScript_CreateIntervalScriptEvent

Description

Use this function to create a Script Event that occurs after a specified amount of time. This type of Script Event can also be repeating so that it will call your script every so many days, hours, minutes, and/or seconds.

Return Type

Text

Format

MMScript_CreateIntervalScriptEvent (**EventName** ; **Script** ; **Interval** ; **DB** ; **EventValue** ; **StartDateTime** ; **Repeat** ; **Persistent** ; **Priority** ; **CurrentScript**)

Required Parameters

EventName

The Name of the Event you are Creating.

Script

The Script to call.

Interval

The amount of time to wait before the Event occurs (Format: "1d 2h 3m 45s").

Optional Parameters

DB

The Database that contains the Script.

EventValue

Any user defined Value you can retrieve when the Event occurs.

StartDateTime

A Date, Time, or Timestamp of when you want the Event to occur the first time.

Repeat

True or False if you want the Event to Repeat until you cancel it.

Persistent

True or False if you want this Event to be saved between FileMaker sessions.

Priority

Defines how interruptive the Event is. (Specify "Low", "Normal", "High", or "Highest".)

Low - The Event occurs if the user has not done anything for 30 seconds or more.

Normal - The Event may occur even if the user has done something within the last 30 seconds, but no script is running.

High - The Event may occur if the user is running a script that is currently paused.

Highest - The Event attempts to run even if there is already a script running.

CurrentScript

Defines what to do with the currently running script (if any). (Specify "Halt", "Exit", "Resume", or "Pause".)

Related Items

MMScript_CancelScriptEvent, MMScript_CreateDateTimeScriptEvent,
MMScript_CreateIdleScriptEvent, MMScript_CreateImmediateScriptEvent,
MMScript_CreateXMLScriptEvent, MMScript_ListScriptEvents

MMScript_CreateXMLScriptEvent

Description

Use this function to create a Script Event using the older XML format. Each of the above functions allow you to create the same type of Script Events you can create with XML here, but if you prefer to use XML or if you are converting older databases that used the SCRIPTit plug-in, you can use this function. See the MMScript_XML entry for documentation on the XML format used with this function.

Return Type

Text

Format

MMScript_CreateXMLScriptEvent (**XML**)

Required Parameters

XML

The XML that defines the Script Event(s).

Related Items

MMScript_CancelScriptEvent, MMScript_CreateDateTimeScriptEvent,
MMScript_CreateIdleScriptEvent, MMScript_CreateImmediateScriptEvent,
MMScript_CreateIntervalScriptEvent, MMScript_ListScriptEvents

MMScript_DisableScriptEvents

Description

Use this function to tell the plug-in to stop processing Script Events.

Return Type

Text

Format

`MMScript_DisableScriptEvents`

Related Items

`MMScript_AreScriptEventsEnabled`, `MMScript_EnableScriptEvents`

MMScript_EnableScriptEvents

Description

Use this function to tell the plug-in to start processing Script Events again if you previously called MMScript_DisableScriptEvents.

Return Type

Text

Format

[MMScript_EnableScriptEvents](#)

Related Items

[MMScript_AreScriptEventsEnabled](#), [MMScript_DisableScriptEvents](#)

MMScript_GetScriptEventName

Description

Use this function to retrieve the name of the Script Event that has called your script. This is useful if you have more than one Script Event calling the same script.

Return Type

Text

Format

`MMScript_GetScriptEventName`

Related Items

`MMScript_GetScriptEventValue`

MMScript_GetScriptEventValue

Description

Use this function to retrieve the Value you assigned to a Script Event. You can also get this Value with the Get(ScriptParameter) calculation function. The Value of a Script Event is useful for passing things like Record IDs to your Script so you know what Record triggered the Script Event.

Return Type

Text

Format

`MMScript_GetScriptEventValue`

Related Items

MMScript_GetScriptEventName, MMScript_UpdateScriptEventValue

MMScript_IsUserIdle

Description

Use this function to determine if the user has been idle for at least a specific amount of time.

Return Type

Number (1=True, 0=False)

Format

MMScript_IsUserIdle (**MinimumSeconds**)

Optional Parameters

MinimumSeconds

The Minimum number of Seconds the User must be Idle for this function to return True. (Default is 30 Seconds.)

Related Items

MMScript_CreateIdleScriptEvent

MMScript_ListScriptEvents

Description

This function returns a list of all currently defined Script Events.

Return Type

Text

Format

[MMScript_ListScriptEvents](#)

Related Items

MMScript_CancelScriptEvent, MMScript_CreateDateTimeScriptEvent,
MMScript_CreateIdleScriptEvent, MMScript_CreateImmediateScriptEvent,
MMScript_CreateIntervalScriptEvent, MMScript_CreateXMLScriptEvent

MMScript_Register

Description

You can use this function to Register the plug-in from a script instead of through the Configuration Dialog. This is useful when the plug-in is being distributed to many computers, allowing you to install and register the plug-in without having to physically visit each computer. This function also allows you to check if the plug-in is already registered or clear the current registration. The plug-in always requires you to accept the License Agreement to use the plug-in. This is usually done by presenting the License Agreement Dialog, but that can be suppressed by using the special option value "I Accept the License Agreement".

Return Type

Text

Format

MMScript_Register (**FirstName** ; **LastName** ; **LicenseKey** ; **Option**)

Required Parameters

FirstName

The Registration First Name you specified when you ordered. (See your Receipt.)

LastName

The Registration Last Name you specified when you ordered. (See your Receipt.)

LicenseKey

The License Key from your Receipt.

Optional Parameters

Option

Specify "Dialog" to enter your Registration Information in a dialog.

Specify "Check" to see if the plug-in is already Registered.

Specify "Clear" to remove the Registration.

Specify "I Accept the License Agreement" to automatically accept the License Agreement dialog without showing it to the end user.

Notes: The "Dialog", "Check", and "Clear" options can also be specified as the first parameter. If you have a Developer License, do not use the "I Accept the License Agreement" value here. See your Developer Instructions file for more information.

Examples

Example 1

Code:

```
MMScript_Register( "My First Name" ; "My Last Name" ; "My License Key" )
```

Result:

Registers the plug-in with the provided registration information (obviously the above is not valid registration information; please see your Receipt).

Example 2

Code:

```
MMScript_Register( "Dialog" )
```

Result:

Displays a dialog for you to enter your First Name, Last Name, and License Key as it appears on your Receipt.

Example 3

Code:

```
MMScript_Register( "Check" )
```

Result:

Returns "Not Registered." or "Registered to <name> for a <license>."

Example 4

Code:

```
MMScript_Register( "My Company Name" ; "My Company Name" ; "My Site License Key" ; "I Acce
```

Result:

Registers the plug-in and uses the "I Accept the License Agreement" option to keep the License Agreement dialog from appearing.

MMScript_ShellExecute

Description

Use this function to call a Command in the System's Command Line Interface. On Mac OS X, this is in the Unix Command Line. On Windows, this is in the DOS Command Line.

Return Type

Text

Format

MMScript_ShellExecute (**Command** ; **WorkingDir**)

Required Parameters

Command

The Command to Execute in the Shell/Command Line Interface.

Optional Parameters

WorkingDir

The Directory to Execute the Command from. (Default is the User's Home folder.)

MMScript_UpdateScriptEventValue

Description

Use this function to Update the Value of an existing Script Event.

Return Type

Text

Format

MMScript_UpdateScriptEventValue (**EventName** ; **NewEventValue**)

Required Parameters

EventName

The Name of the Event you are Updating.

NewEventValue

The New Value for the Event.

Related Items

MMScript_GetScriptEventValue

MMScript_Version

Description

This function returns the current version of MMScript. This function is useful for testing whether or not the plug-in is installed and enabled. If you call this function and a question mark ("?.") is returned, then the plug-in is either not installed or not enabled.

Return Type

Text

Format

`MMScript_Version`

Related Items

MMScript_VersionAutoUpdate

Example

Code:

```
MMScript_Version
```

Result:

Returns the MMScript version like "MMScript v.1.1.2".

MMScript_VersionAutoUpdate

Description

This function returns an Auto Update friendly Version number of MMScript. The format of this version number is always exactly 8 digits long. The first two digits represent the major version of the plug-in (zero-filled). The third and fourth digits represent the minor version of the plug-in (zero-filled). The fifth and sixth digits represent the update portion of the version (zero-filled). The final two digits represent a special build number or a beta version number and will usually be zeros.

As an example, for MMScript 1.1.2, the major version is 1, the minor version is 1, the update number is 2, and there is no special build or beta version defined. So, the resulting Auto Update friendly version number would be 01010200.

Return Type

Number

Format

`MMScript_VersionAutoUpdate`

Related Items

MMScript_Version

Example

Code:

```
MMScript_VersionAutoUpdate
```

Result:

Returns 01010200 for MMScript version 1.1.2.

