



Troi Activator Plug-in 1.5 USER GUIDE

January 2006



Troi Automatisering

Vuurlaan 18

2408 NB Alphen a/d Rijn

The Netherlands

Telephone: +31-172-426 606

Fax: +31-172-470 539

You can also visit the Troi web site at: <<http://www.troi.com/>> for additional information.

Troi Activator Plug-in is copyright 2000-2006 of Troi Automatisering. All rights reserved

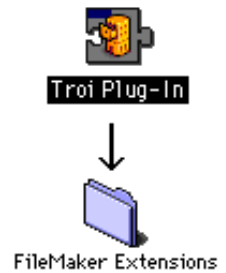
Table of Contents

Installing plug-ins.....	1
If you have problems.....	1
What can this plug-in do?.....	2
Software Requirements.....	2
Getting started	2
Using external functions.....	2
Where to add the external functions?.....	3
Simple example.....	3
Triggering Scripts on other computers	4
Think about Security first!	3
Start listening	3
Sending trigger messages	4
Summary of functions.....	5
Function Reference.....	6
Actr-Control	6
Actr-DeleteEvent	8
Actr-GetEventInfo	9
Actr-GetIPAddress	10
Actr-ScheduleEvent	11
Actr-SendRemoteEvent	13
Actr-Sleep	15
Actr-StartListener	17
Actr-StopListener	19
Actr-Version	20

Installing plug-ins

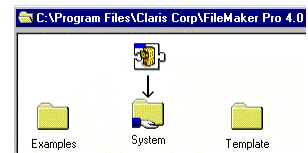
For MacOS:

- Quit FileMaker Pro.
- Put the file "Activator Plug-in" from the folder "Mac OS Plug-in" into the "FileMaker Extensions" folder in the FileMaker Pro application folder.
- If you have installed previous versions of this plug-in, you are asked: "An older item named "Activator Plug-in" already exists in this location. Do you want to replace it with the one you're moving?". Press the OK button.
- Start FileMaker Pro. The first time the Activator Plug-in is used it will display a flash dialog box, indicating that it is loading and showing the registration status.



For Windows:

- Quit FileMaker Pro.
- Put the file "activatr.fmx" from the directory "Windows" into the "System" subdirectory in the FileMaker Pro application directory.
- If you have installed previous versions of this plug-in, you are asked: "This folder already contains a file called 'activatr.fmx'. Would you like to replace the existing file with this one?". Press the Yes button.
- Start FileMaker Pro. The Troi Activator Plug-in will display a dialog box, indicating that it is loading and showing the registration status.



TIP You can check which plug-ins you have loaded by going to the plug-in preferences: Choose **Preferences** from the **Edit** menu, and then choose **Plug-ins**.

You can now open the file "All Activator Examples.fp5" to see how to use the plug-in's functions. There is also a function overview available.

If you have problems

This user guide tries to give you all the information necessary to use this plug-in. So if you have a problem please read this user guide first. Also you might visit our support web page:

<<http://www.troi.com/support/>>

This page contains FAQ's (Frequently Asked Questions), help on registration and much more. If that doesn't help you can get free support by email. Send your questions to **support@troi.com** with a full explanation of the problem. Also give as much relevant information (version of the plug-in, which platform, version of the operating system, version of FileMaker Pro) as possible.

If you find any mistake in this manual or have a suggestion please let us know. We appreciate your feedback!

TIP You can get more information on returned error codes from our OSerrrs database on our web site: <<http://www.troi.com/software/oserrrs.html>>. This free FileMaker database lists all error codes for Windows and Mac OS!

What can this plug-in do?

The Troi Activator Plug-in is a very powerful tool for triggering scripts. All from within FileMaker you can:

- trigger a script on a specified date and time
- schedule events which trigger any script you want
- validate fields with a script
- notify a different user on a different computer of changes with the click of a button
- put the computer to sleep till it needs to be activated again (Mac OS 8.x-9.x only)
- and more...

Software requirements

The Troi Activator Plug-in will work with FileMaker Pro 4.0 or later on all platforms (Windows 95/98/Me, Windows NT/2000/XP, Mac OS 8.x, Mac OS 9.x, Mac OS X 10.0.1* or later). It will also work with a bound runtime, created by FileMaker Developer Edition 4.0 and FileMaker Developer 5. You can also use FileMaker Server to serve databases that use functions of the Activator plug-in. You need to have the plug-in installed at the clients that use these functions.

This means that in Troi Activator Plug-in version 1.3 earlier platform limitations are removed.

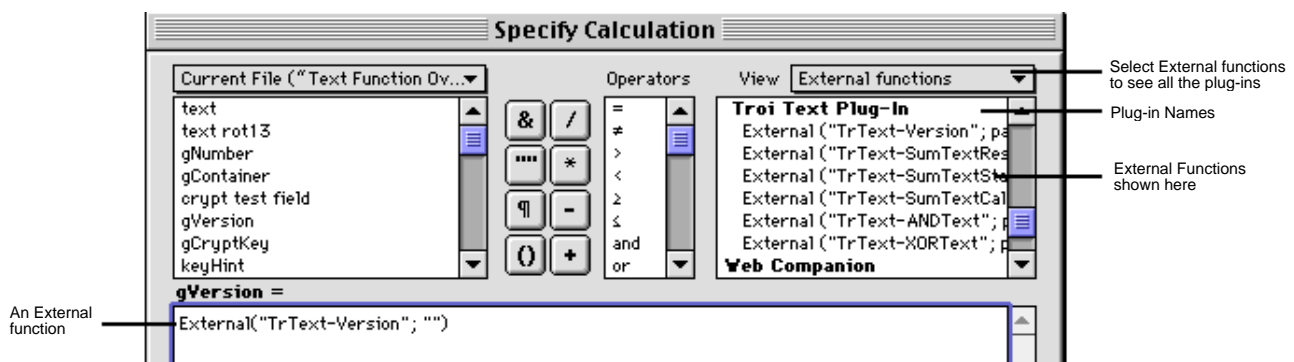
* Actr-Sleep is not available on Windows and Mac OS X 10.0.4 and earlier. A future update of Mac OS X may implement this functionality without the need for updating the Activator.

NOTE We have done only limited testing for FileMaker Pro 4.0, FileMaker Pro 4.1 and runtimes from FileMaker Developer Edition 4.0 and on Mac 68k machines and found no significant problems. However we are dropping support for these FileMaker versions to be able to concentrate on current and future versions of FileMaker.

Getting started

Using external functions

The Activator Plug-in adds new functions to the standard functions that are available in FileMaker Pro. The functions added by a plug-in are called external functions. You can see those extra functions for all plug-ins at the top right of the Specify Calculation box:



You use special syntax with external functions: `External("function name",parameter)` where function name is the name of an external function. The parameter is required, even if it's only `""`. Plug-ins don't work directly after installation. To access a plug-in function, you need to add the calls to the function in a calculation for example in a text calculation in Define Fields or in a ScriptMaker Script.

IMPORTANT In the United States, commas act as list separators in functions. In other countries semicolons might be used as list separators. The separator being used depends on the operating system your computer uses, as well as the separator used when the file was created. All examples show the functions with commas. For example: `External("Actr-Version", "")` will become `External("Actr-Version"; "")` in such a file.

Where to add the external functions?

External functions for this plug-in are intended to be used in a script step using a calculation. For most functions of this plug-in it makes no sense to add them to a define field calculation, as the functions will have side effects. The function "Actr-ScheduleEvent" of this plug-in can be used in a validation calculation when you are defining fields (choose Define from the File menu). See the Validation.fp5 example.

Simple example

We start with a simple example to get you started. Say you have a database Remote.fp5, with a global text field called gMyIPAddress, in which you are going to store the IP address of the computer. Create a new ScriptMaker Script called "Get my IP address". Then add the following script step to a script:

```
Set Field[gMyIPAddress, External("Actr-GetIPAddress", "")]
```

This will put the IP address of your computer into field gMyIPAddress; for example :

192.123.234.12

NOTE Function names, like "Actr-GetIPAddress" are case sensitive. Be sure to spell them right, or get them from the External Functions list at the top right of the "Specify Calculation" dialog.

Please take a close look at the included example files, as they provide a great starting point. From there you can move on, using the functions of the plug-in as building blocks. Together they give you all the tools you need to trigger scripts from anywhere!

Triggering Scripts on other computers

First a few definitions: The receiver is the computer where a script must be triggered. The sender is the computer which sends a script trigger message to the receiver. Any computer can be a receiver and sender at the same time.

Think about Security first!

Before implementing we advice you to think about the security issues, as triggering scripts is a very powerful feature. You do not want to have a "Delete All Records" script be triggered by an anonymous hacker. See below how you can limit security risks of remote triggering.

Start listening

To be able to trigger scripts on the receiver computer it must first start listening for trigger messages. If you

don't want the Activator Plug-in to trigger scripts initiated by other computers you should not start listening. The plug-in will then ignore all remote trigger messages. In ScriptMaker add the following script step:

```
Set Field [ gErrorCode, External("Actr-StartListener",  
                                "-DefaultPortNumber|" & Status(CurrentFileName) & "|12345") ]
```

This will start the listening for messages on the default port (UDP port 54242). Only messages which have securityID 12345 are handled, other messages will be ignored. This makes listening for messages much safer. No one can trigger a script on your computer, unless they know the securityID!

TIP For the securityID you can use anything you like. For example a large random number, a password or a combination of this. If you specify an empty securityID all trigger messages will be executed, even if the sender has included a non-empty securityID. See the example file "Remote.fp5" for a random number algorithm.

The receiver needs to tell other computers that it is listening for messages. A sender needs to know the IP address, the port number and the securityID (if used). A convenient way to distribute this information is via a shared FileMaker database, in which a receiver puts the needed information. See the example file Remote.fp5 where this is implemented.

NOTE This is just a suggested implementation for getting this information to the sender. It is not necessary to have a shared database for remote triggering to work. You can implement this in a different way, for example by publishing it in on the web or sending email. The information might even be always the same, so you don't need to communicate it.

TIP2 If the receiver is behind a firewall, the port on which the receiver is listening must be opened. If the receiver started listening on the default port this is UDP port 54242. See your system administrator for this.

Sending trigger messages

To send trigger scripts you only have to send the message to the receiver computer.

We assume that in your FileMaker file "Remote.fp5" the following fields are defined:

<u>Field name</u>	<u>Type</u>	<u>Possible contents</u>
Name	Text	Peter Baanen
IP Address	Text	192.1.123.24
SecurityID	Text	secretpassword
FileName	Text	Remote.fp5
ScriptName	Text	TriggerScript1
gYourID	Global, text	123
gYourText	Global, text	Please look at this record.
gErrorCode	Global, text	0

In ScriptMaker add the following script step:

```
Set Field [ gErrorCode, External("Actr-SendRemoteEvent", "-PortNumber =51000|" &  
                                IP Address & "|" & SecurityID & "|" & FileName & "|" &  
                                ScriptName & "|" & gYourID & "|" & gYourText) ]
```

This will send the message to the receiver at the specified IP address and port 51000. The receiver must be listening on this port and also the securityID must be the same. You can use the global field gYourID and gYourText to send information to the receiver, for example a record ID and a message.

Summary of functions

The Troi Activator Plug-in adds the following functions:

<u>function name</u>	<u>short description</u>
Actr-DeleteEvent	remove an event from the memory of the plug-in.
Actr-GetEventInfo	return information associated with the event, like an ID and/or text.
Actr-GetIPAddress	get the IP address of this computer.
Actr-ScheduleEvent	schedule a script to be automatically triggered in the future.
Actr-SendRemoteEvent	send a trigger script message to a remote computer.
Actr-Sleep	put the computer into sleep and let it wake up again at a specified time.
Actr-StartListener	start listening for messages from other computers.
Actr-StopListener	stop listening for messages from other computers.
Actr-Version	determine which version of the plug-in is loaded; also used for registration.

Function Reference

Actr-Control

Syntax Set Field [result, External("Actr-Control", "switches |password")]

Controls the triggering of the plug-in. You can disable and enable local triggering of the plug-in. This allows you to change the contents of a field which is validated by triggering a script.

Parameters

switches	these determine how the function works. You can disable or enable all functions
password	the password to be used

switches can be one of the following:

-DisableLocalTriggers	disable triggering of scripts on this computer and also the scheduling of new triggers.
-EnableLocalTriggers	enable triggering of scripts on this computer and also the scheduling of new triggers.
-GetLocalTriggerStatus	get the status of the local triggering: 1 is on 0 off.

Returned result

The returned result is an error code. An error always starts with 2 dollars, followed by the error code. You should always check for errors. Returned error codes can be:

0	no error
\$\$-4217 pwdAlreadySet	password already set (enable first)
\$\$-4218 alreadyEnabled	functions are already enabled
\$\$-4219 pwdWrong	password was wrong

Other errors may be returned.

Special considerations

When you call Actr-ScheduleEvent when it is disabled an error code of -4220 is returned.

When you disable triggering also the scheduling of new local trigger events is disabled until restart of FileMaker. Other users can still trigger scripts on their computer and you can also send a remote event to another computer.

Example usage

```
Set Field[result, External("Actr-Control", "-DisableLocalTriggers |secret password")]
```

This will disable the trigger functions of the plug-in on this computer.

Example 2

We assume that in your FileMaker file the following fields are defined:

gErrorCode	Global, number
gPassword	Global, text

Actr-Control

gPassword should contain a password, for example "rapunsel". In ScriptMaker add the following to the steps to a script:

```
Set Field[gErrorCode, External("Actr-Control", "-DisableLocalTriggers|" & gPassword)]  
#... do for example a Replace function on a script validated function...  
Replace Contents[ValidateField , newValue]  
Set Field[gErrorCode, External("Actr-Control", "-EnableLocalTriggers|" & gPassword)]
```

This will first disable the plug-ins triggers. Then you can change the values. The last step reactivates triggering.
This will temporary enable the plug-in functions.

Actr-DeleteEvent

Syntax Set Field [result, External("Actr-DeleteEvent", "switches |eventID")]

This function will remove an event from the memory of the plug-in.

Parameters

switches	determines which information is to be returned
eventID	(optional) the event to delete

switches can be one of this:

-DeleteLastTriggered	delete the event that was last triggered
-DeleteAllTriggeredEvents	delete all triggered events
-DeleteAllEvents	delete all events
-DeleteByID	delete the event specified in the next parameter

Returned result

If successful it returns 0. If unsuccessful it returns an error code starting with \$\$ and the error code. Possible codes are:

0 = no error

\$\$-50 = parameter error, check if your parameters are correct.

\$\$-41 = not enough memory

Other errors may be returned.

Special considerations

You can also delete events when you are retrieving data from the event. See the function: Actr-GetEventInfo. for this.

Example usage

Set Field[result, External("Actr-DeleteEvent", "-DeleteAllTriggeredEvents")]

This command will remove all triggered events.

Example 2

We assume that in your FileMaker file the following fields are defined:

gErrorCode	Global, text
gEventID	Global, number

gEventID contains an eventID. In ScriptMaker add the following script step to your Trigger script:

Set Field[gErrorCode, External("Actr-DeleteEvent", "-DeleteByID" & "|" & gEventID)]

This will remove the event with the same EventID as in field gEventID from the memory of the plug-in.

Actr-GetEventInfo

Syntax Set Field [result, External("Actr-GetEventInfo", "switches|eventID")]

This function will return information that was previously associated with the event, like yourID and yourText.

Parameters

switches determines which information is to be returned
eventID (optional) the event to retrieve info from

switches must contain only one of this:

-GetYourID retrieves the yourID data
-GetYourText retrieves the yourText data
-GetActivatorEventID retrieves the internal EventID

and also switches must contain one of this:

-LastTriggered retrieves the information from the event that was triggered last
-ByEventID retrieves the information from the event with the eventID in the next parameter
-FirstSilentEvent retrieves the information from the first silent event

optional switches can contain this switch:

-DeleteThisEvent delete the event after the data is returned

Returned result

If successful it returns 0.

If unsuccessful it returns an error code starting with \$\$ and the error code. Possible codes are:

0 = no error

\$\$-50 = parameter error, check if the supplied parameters are correct.

\$\$-41 = not enough memory

\$\$-1 = non existing Event

Other errors may be returned.

Example usage

Set Field[result, External("Actr-GetEventInfo", "-LastTriggered -GetYourID -DeleteThisEvent")]

This command will return the YourID data for the last event triggered. The event data will be deleted.

Example 2

We assume that in your FileMaker file the following fields are defined:

gEventID Global, number
gEventText Global, text

gEventID contains the eventID. In ScriptMaker add the following script step to your Trigger script:

Set Field[gEventText, External("Actr-GetEventInfo", "-ByEventID -GetYourID" & "|" & gEventID)]

This command will fill the field gEventText with the text that was supplied earlier, when the event was created.

Actr-GetIPAddress

Syntax Set Field [result , External("Actr-GetIPAddress", "switches")]

Gets the IP address(es) of this computer.

Parameters

switches You can leave this blank or put one of this:

- InterfaceIndex = x get the IP address of interface x, the first interfaces starts at 1
- Defaultinterface get the default IP address of this computer

Returned result

the IP address of this computer.

Special considerations

To publish your address you can put the returned IP address in a shared database, together with other clarifying data, like your name. This makes it easy to send a remote event to a particular person: Look up the IP address that is in the same record as the name. See the Remote.fp3 example file, where this is worked out.

NOTE: On computers with multiple IP addresses GetIPAddress returns the first IP address if you leave the switch empty. On Mac OS you can (starting with version 1.3.2) also ask for other than the first IP address.

Example usage

```
Set Field[result, External("Actr-GetIPAddress", "") ]
```

This command will return the IP address for this computer, for example "192.168.1.24".

Example 2

```
Set Field[result, External("Actr-GetIPAddress", "-InterfaceIndex = 1") ]  
Set Field[result, External("Actr-GetIPAddress", "-InterfaceIndex = 2") ]
```

This command will return the first and the second IP address for this computer, for example "192.168.1.24" and "198.123.32.1".

Actr-ScheduleEvent

Example 2

We assume that in your FileMaker file the following fields are defined:

gTriggerDate	Global, date
gTriggerTime	Global, time
gEventID	Global, number

gTriggerDate and gTriggerTime can be filled with the date and time the script should be triggered. In ScriptMaker add the following script step:

```
Set Field[gEventID, External("Actr-ScheduleEvent", "-AddSingleEvent|" &
    gTriggerDate & "|" & gTriggerTime & "|" & Status(CurrentFileName) & "|TriggerScript|" & Status
(CurrentRecordID) & "|Check record please.")]
```

This command will trigger the script "TriggerScript" in the current file. It will be triggered on the day and time that are in the global fields gTriggerDate and gTime.

Actr-SendRemoteEvent

from FileMaker Developer 4 , 5.x and 6 on all platforms.

Example usage

```
Set Field [ gErrorCode, External("Actr-SendRemoteEvent",  
    "-DefaultPortnumber|192.168.1.1|secretword|Filename.fp5|Triggerscript1|1234567|Hello there!") ]
```

Example 2

We assume that in your FileMaker file the following fields are defined:

gIPAddress	Global, text
gSecurityID	Global, text
gYourID	Global, text
gYourText	Global, text
gErrorCode	Global, text

gIPAddress and gSecurityID should contain the IP address resp. the SecurityID of the computer you want to trigger.
gYourID and gYourText can be filled with the message you want to send to the other computer. In ScriptMaker add the following script step:

```
Set Field [gErrorCode, External("Actr-SendRemoteEvent",  
    "-PortNumber =51000|" & gIPAddress & "|" & gSecurityID & "|" & "Remote.fp3" & "|" & "Triggerscript1" &  
    "|" & gYourID & "|" & gYourText) ]
```

This will trigger script "Triggerscript1" in database "Remote.fp3" on the remote computer.

Actr-Sleep

Syntax Set Field [result, External("Actr-Sleep", "switches |date|time")]

Put the computer into sleep mode (if present). You can specify when to wake up again.

Parameters

switches	set to "-TillDateTime" (in the future new switches may be added)
date	the date the computer must wake up
time	the time the computer must wake up

Returned result

The returned result is an error code. If successful it returns 0. An error always starts with 2 dollars, followed by the error code. You should always check for errors. Returned error codes can be:

0 = no error

\$\$-50 = parameter error, check if your parameters are correct

\$\$-4121 = this function is not available for this Operating System (windows)

\$\$-4126 = the date specified should be in the future (kErrDateInPast)

\$\$-4131 = this computer does not have suitable hardware to do this

Other errors may be returned.

Special considerations

- This is a Mac OS only function. It works for Mac OS 8.X-9.X on computers that support sleep.

The current release of Mac OS X (10.0.4) does not implement this API. This may be fixed in an update.

- The function returns before the computer has gone into sleep. Use a Pause/Resume step with 20 seconds pause, to be sure the next step is executed after the sleep.

- Waking up starts at the specified time. It might take a few seconds for the computer to be completely awake. If it is critical that the computer is awake at a specified time, subtract 30 seconds from the wake time.

Example usage

Set Field[result, External("Actr-Sleep", "-tilldatetime|730485|1000")]

This command will put the computer into sleep and wake it up on day 730485, this date is equivalent to Dec 31, 2000. It will wake up 1000 seconds after midnight. Note that this example is somewhat simplified, normally you don't need to know about the date and time formatting (See also example 2)

Example 2

We assume that in your FileMaker file the following fields are defined:

gWakeDate	Global, date
gWakeTime	Global, time
result	Global, text

Actr-Sleep

gWakeDate and gWakeTime can be filled with the date and time the computer must wake up. In ScriptMaker add the following script step:

```
Set Field[result, External("Actr-Sleep", "-TillDateTime |" & gWakeDate &"|" & gWakeTime)]
```

This command will put the computer to sleep and wake the computer on the day and time that are in the global fields "gWakeDate" and "gWakeTime".

Actr-StartListener

Syntax Field [result , External("Actr-StartListener", "switches|currentFileName|securityID")]

Start listening for messages from other computers.

Parameters

switches	determines the behaviour of this command
currentFileName	the name of the file that contains the script that must be triggered later
securityID	(optional) a security ID which the sender needs to add to sent messages

switches must contain only one of this:

-Defaultportnumber	use the default port number of the Activator (UDP port 54242)
-Portnumber =<portnumber>	use the specified portnumber

Returned result

If successful it returns 0.

If unsuccessful it returns an error code starting with \$\$ and the error code. Possible codes are:

0 = no error

\$\$-50 = parameter error. There was an error with the parameters.

\$\$-4215 = invalid FileMaker Version. On windows you need FileMaker 5 or later.

\$\$-4227 = already listening.

\$\$-4228 = portnumber out of range: use 0 to 65535.

Special considerations

Which port you should choose is dependent on your network situation: You should use a port number that is not in use. Try to use the default port number. If this port is occupied choose a number in the private range: 49152 - 65535.

Below you find how ports are currently assigned:

0 - 1023:	Well Known Ports, used by standard protocols. Don't use for Activator.
1024 - 4915:	Registered Ports. Not recommended for Activator.
49152 - 65535:	Dynamic and/or Private Ports.

The securityID makes listening for messages safe. No one can trigger a script on your computer, unless they know the securityID. If the sender sends a different securityID no triggering occurs!

For the securityID you can for example use (random) numbers or a password, and you can distribute it in a field of a shared database. See the example file "Remote.fp3". If you specify an empty securityID all messages will be triggered, even if the sender has included a non-empty securityID.

NOTE Starting with version 1.3 this function now works for all version of FileMaker Pro 4 and later and bound runtimes from FileMaker Developer 4 and 5 on all platforms.

Example usage

Set Field [gErrorCode, External("Actr-StartListener", "-defaultportnumber|" & Status(CurrentFileName) & "|12345")]

This will start the listening for messages. Only messages which have securityID 12345 are handled, others will be ignored.

Example 2

Actr-StartListener

We assume that in your FileMaker file the following fields are defined:

gSecurityID	Global, text
gErrorCode	Global, text

gSecurityID should contain the security ID, for example "secretpassword". In ScriptMaker add the following script step:

```
Set Field [ gErrorCode, External("Actr-StartListener", "-portnumber=50505|" & Status(CurrentFileName) & "|" & gSecurityID) ]
```

This will start the listening for messages on port 50505. Only messages which are sent to this port and have securityID that's equal to the value in field gSecurityID are handled, others are ignored.

Actr-StopListener

Syntax Field [result , External("Actr-StopListener", "switches")]

Stops listening for messages from other computers.

Parameters

switches not used, reserved for future use. Leave blank or put "-unused"

Returned result

If successful it returns 0.

If unsuccessful it returns an error code starting with \$\$ and the error code.

Special considerations

At the moment this function only returns 0.

Example usage

Set Field [gErrorCode, External("Actr-StopListener", "")]

Actr-Version

Syntax Set Field [result, External("Actr-Version", "switches")]

Use this function to see which version of the plug-in is loaded.

Note: This function is also used to register the plug-in.

Parameters

switches determine the behaviour of the function

switches can be one of this:

- GetString the version string is returned (default)
- GetVersionNumber Returns the version number of the plug-in
- ShowFlashDialog Shows the Flash Dialog of the plug-in (returns 0)

If you leave the parameter empty the version string is returned.

Returned result

The function returns "" if this plug-in is not loaded. If the plug-in is loaded the result depends on the input parameter. It is either a:

VersionString:

If you asked for the version string it will return for example "Troi Activator Plug-in 1.4"

VersionNumber:

If you asked for the version number it returns the version number of the plug-in x1000. For example version 1.4 will return number 1400.

ShowFlashDialogResult:

This will show the flash dialog and then return the error code 0.

Special considerations

IMPORTANT Always use this function to determine if the plug-in is loaded. If the plug-in is not loaded use of external functions may result in data loss, as FileMaker will return an empty field to any external function that is not loaded.

Example usage

External("Actr-Version", "") will for example return "Activator Plug-in 1.4"

Example 2

External("Actr-Version", "-GetVersionNumber") will return 1100 for version 1.1.

External("Actr-Version", "-GetVersionNumber") will return 1401 for version 1.4b1

External("Actr-Version", "-GetVersionNumber") will return 2130 for version 2.1.3

So for example to use a feature introduced with version 1.4 test if the result is equal or greater than 1400.