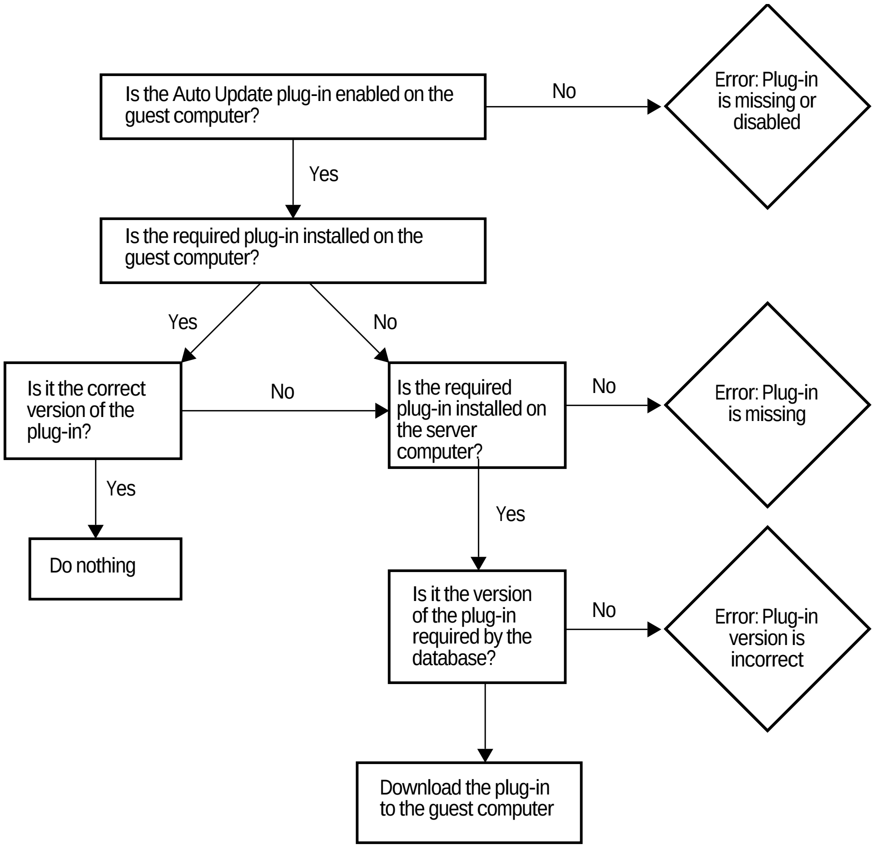
Updating plug-ins automatically

If you use plug-ins in your databases to extend the functionality of FileMaker Pro® 5.5, you can use the Auto Update feature to ensure database guests always have the most current plug-in software installed on their computers.

When you distribute FileMaker Pro databases in a server/guest environment, any plug-ins required by your database can be installed on each guest computer that accesses the database. If you make a change to a plug-in, the updated plug-in should be distributed to all guest computers that access the database. In a large organization, manually updating many guest computers can be time consuming. You can use the Auto Update feature, available in FileMaker Server 5.5, to download updated files automatically.

The following illustration shows one way to use Auto Update to check both the guest and server computers for the existence of a plug-in required by your database.



Is the Auto Update plug-in enabled on the guest computer?
No
Error: Plug-in is missing or disabled
Yes
Is the required plug-in installed on the guest computer?
Yes
No
Is it the correct version of the plug-in?
No
Is the required plug-in installed on the server computer?
No
Error: Plug-in is missing
Yes
Do nothing
Yes
Is it the version of the plug-in required by the database?
No
Error: Plug-in version is incorrect
Download the plug-in to the guest computer

For automatic update to work properly, you must:

- Enable the Auto Update plug-in in FileMaker Pro 5.5 and set up your database to call FileMaker Pro 5.5 external functions. These functions verify the existence and version of required plug-ins on both the guest and server computers and download plug-ins, if needed.
- Give plug-ins and plug-in version files to the FileMaker Server administrator, so they can be placed in the correct location on the server computer and downloaded as needed. Remind the server administrator to turn on Auto Update in FileMaker Server. This feature can be enabled in the FileMaker Server Properties (Windows) or Preferences (Mac OS) or UseAutoUpdate preference parameter (Linux) in the FileMaker Server configuration file (/etc/fmserver.conf).

How automatic downloading works

Auto Update ensures guests have current plug-ins by addressing two situations:

- The guest is opening your file for the first time and the plug-ins it requires don’t exist on the guest computer.
- The guest has opened your file previously, but has an outdated version of a required plug-in and needs an update.

The following sections describe the general sequence of events when a plug-in is missing from or out-of-date on the guest computer.

When the required plug-in is missing from the guest computer

The guest starts FileMaker Pro 5.5 and attempts to open your database hosted by FileMaker Server 5.5. Scripts, which you have defined in your database, run the external functions for Auto Update in the following order:

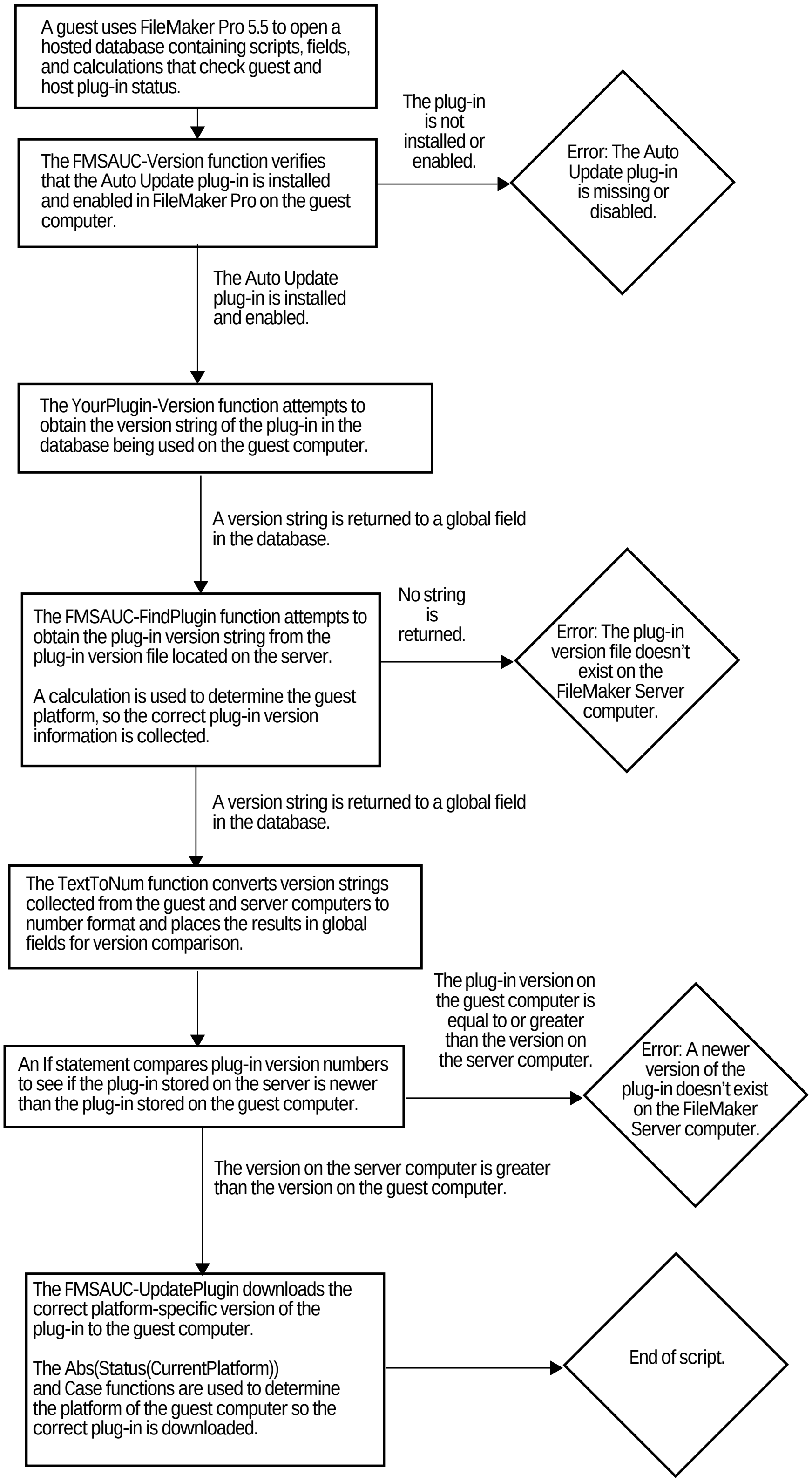
1. The external function FMSAUC-Version runs, which returns the name and version of the Auto Update plug-in available in FileMaker Pro 5.5. If plug-in information isn’t returned, it is assumed that the plug-in is missing from or disabled on the guest computer.
2. The external function YourPlugin-Version runs, which returns the name and version of the plug-in required by your database. *YourPlugin-Version* is the external function Version for the plug-in used in your database. If plug-in version information is not returned, it is assumed that the plug-in is missing or disabled on the guest computer.
3. The external function FMSAUC-FindPlugin runs, which searches the AutoUpdate folder on the server for a text file containing the version of the plug-in stored there, and returns the version string.
4. The version from the server can be compared to the version from the guest computer collected by the YourPlugin-Version function. In this case, no version information was returned by YourPlugin-Version, so the version on the server is greater than the version of the plug-in on the guest computer.
5. The external function FMSAUC-UpdatePlugin runs, placing the required plug-in in the System folder (Windows) or FileMaker Extensions folder (Mac OS) on the guest computer.

When the required plug-in is out of date on the guest computer

The guest starts FileMaker Pro 5.5 and attempts to open your database hosted by FileMaker Server 5.5. Scripts, which you have defined in your database, run the external functions for Auto Update in the following order:

1. The external function FMSAUC-Version runs, which returns the name and version of the Auto Update plug-in available in FileMaker Pro 5.5. If plug-in information isn’t returned, it is assumed that the plug-in is missing from or disabled on the guest computer.
2. The external function YourPlugin-Version runs, which returns the name and version of the plug-in required by your database. If plug-in version information is not returned, it is assumed that the plug-in is missing or disabled on the guest computer.
3. The external function FMSAUC-FindPlugin runs, which searches the AutoUpdate folder on the server for a text file containing the version of the plug-in stored there, and returns the version string.
4. The version from the server can be compared to the version from the guest computer collected by the YourPlugin-Version function. In this case, the version on the server is newer than the version on the guest computer, so the version on the server is greater than the version of the plug-in required by your database.
5. The external function FMSAUC-UpdatePlugin runs, placing the updated plug-in in the System folder (Windows) or FileMaker Extensions folder (Mac OS) on the guest computer.

The following illustration shows how you can use the external functions provided by the Auto Update plug-in with a few standard FileMaker Pro 5.5 functions to check first the guest computer and then the server computer (if necessary) for a plug-in required by your database.


A guest uses FileMaker Pro 5.5 to open a hosted database containing scripts, fields, and calculations that check guest and host plug-in status.
The FMSAUC-Version function verifies that the Auto Update plug-in is installed and enabled in FileMaker Pro on the guest computer.
The plug-in is not installed or enabled.
Error: The Auto Update plug-in is missing or disabled.
The Auto Update plug-in is installed and enabled.
The YourPlugin-Version function attempts to obtain the version string of the plug-in in the database being used on the guest computer.
A version string is returned to a global field in the database.
The FMSAUC-FindPlugin function attempts to obtain the plug-in version string from the plug-in version file located on the server.
A calculation is used to determine the guest platform, so the correct plug-in version information is collected.
No string is returned.
Error: The plug-in version file doesn't exist on the FileMaker Server computer.
A version string is returned to a global field in the database.
The TextToNum function converts version strings collected from the guest and server computers to number format and places the results in global fields for version comparison.
An If statement compares plug-in version numbers to see if the plug-in stored on the server is newer than the plug-in stored on the guest computer.
The plug-in version on the guest computer is equal to or greater than the version on the server computer.
Error: A newer version of the plug-in doesn't exist on the FileMaker Server computer.
The version on the server computer is greater than the version on the guest computer.
The FMSAUC-UpdatePlugin downloads the correct platform-specific version of the plug-in to the guest computer.
The Abs(Status(CurrentPlatform)) and Case functions are used to determine the platform of the guest computer so the correct plug-in is downloaded.
End of script.

Setting up Auto Update in your database

There are several ways to set up Auto Update in your database. This document describes one way, which uses simple script steps and global fields, and is based on the sample file named AutoUpdatePlugin.fp5 installed with FileMaker Server 5.5. This sample file is also available on the FileMaker, Inc. web site.

Note This procedure assumes that you know how to define fields and scripts and use plug-ins in FileMaker Pro database files.

Overview

To set up plug-in version checking in your database, you need to write a few simple script steps that run when a guest opens your database. The first directs FileMaker Pro to a start-up layout, which contains global fields that store version information about your database's plug-in requirements. This script calls a script named Run Plug-in Scripts, which calls sub-scripts that check for the required plug-in on both the guest and server computers, collects version numbers of plug-ins that exist in these locations, compares them, and downloads an updated plug-in from the server, if needed.

In addition, you also need to produce a version information file to be stored on the server computer, and you might need a MacBinary version of the plug-in itself. These files are produced in the sample file using two additional scripts.

To set up plug-in version checking in your database

1. Open your database using FileMaker Pro 5.5 and enable the Auto Update plug-in in the Application Preferences dialog box.
2. Create a layout to contain:
   - global fields to collect plug-in version information or result codes
   - buttons to manually run scripts that create a plug-in version information file and MacBinary plug-in file (if needed) for storage on the server

   For an example layout that contains all the fields and buttons you need, see the STARTUP layout in the AutoUpdatePlugin.fp5 sample file.
3. Write a script that uses the YourPlugin-Version function to check the version of the plug-in on the guest computer and place version information in the global field named Local_Version in your Auto Update layout.

   For script syntax, see the Local Plug-in Check script in the AutoUpdatePlugin.fp5 sample file. Information about the YourPlugin-Version function should be included in the third-party plug-in documentation.
4. Write a script that uses the FMSAUC-FindPlugin function to check the version of the plug-in on the server computer and place version information the global field named Remote_Version in your Auto Update layout.

   For script syntax, see the Remote Plug-in Check script in the AutoUpdatePlugin.fp5 sample file. For information about the FMSAUC-FindPlugin function, see "FMSAUC- FindPlugin" on page 6.
5. Write a script that converts version information to number format and places the result in additional global fields in your Auto Update layout.

   This conversion to number format is required for a comparison of the plug-in version information collected from the guest and server computers.

   For script syntax, see the Get Version Numbers script in the AutoUpdatePlugin.fp5 sample file. For information about the TextToNum function, see the FileMaker Pro onscreen Help.
6. Write a script to download the plug-in from the server computer, if the version on the guest computer doesn't exist or is outdated.

   For script syntax, see the Download Plug-In script in the AutoUpdatePlugin.fp5 sample file. For information about the FMSAUC-UpdatePlugin function, see "FMSAUC-UpdatePlugin" on page 7.

7. Write a “parent” script to:
   - make sure the Auto Update plug-in exists and is enabled on the guest computer
   - run the individual version checking and downloading scripts you wrote earlier

   For script syntax, see the Run Plug-in Scripts script in the AutoUpdatePlugin.fp5 sample file. For information about the FMSAUC-Version function, see “FMSAUC-Version” on page 6.
8. Write a script to create a plug-in version file to give to the FileMaker Server administrator for storage on the server computer. Then define a button in your Auto Update layout to perform the script.

   For script syntax, see the CreatePVF script in the AutoUpdatePlugin.fp5 sample file. For information about the FMSAUC-SaveVersion function, see “FMSAUC-SaveVersion” on page 8.
9. If you expect Mac OS guests to use your database, write a script to create a MacBinary version of the plug-in to give to the FileMaker Server administrator for storage on the server computer. Then define a button in your Auto Update layout to perform the script.

   For script syntax, see the CreateMacBin script in the AutoUpdatePlugin.fp5 sample file. For information about the FMSAUC-SaveAsMacBin function, see “FMSAUC-SaveAsMacBin (Mac OS)” on page 9.

External function reference

Auto Update consists of the five external functions summarized in the following table. Detailed descriptions of each function are located in the reference sections that follow.

Note These functions are available in FileMaker Pro 5.5 only if the Auto Update plug-in is installed on your computer and enabled in the Plug-ins tab of Application Preferences.

This function	Does this
FMSAUC-Version	Returns the name and version of the Auto Update plug-in that exists in the System (Windows) or FileMaker Extensions (Mac OS) folder on the guest computer. If no version is returned, FileMaker Pro assumes that the plug-in is missing or isn’t enabled on the guest computer.
FMSAUC-FindPlugin	Returns the plug-in version string from the version text file (.TXT) stored in the AutoUpdate folder on the server computer. If no string is returned, FileMaker Pro assumes that the plug-in doesn’t exist on the server computer.
FMSAUC-UpdatePlugin	Downloads a plug-in file from the Auto Update folder on the server computer or returns an error code if the file can’t be downloaded.
FMSAUC-SaveVersion	Saves plug-in version information in a text file for storage on the server.
FMSAUC-SaveAsMacBin (Mac OS)	Converts a plug-in file to MacBinary format. This conversion is necessary if guest computers are running the Mac OS.

The syntax for an external function is:

```
External (name, parameter)
```

FMSAUC-Version

Format

`External (“FMSAUC-Version”, 0)`

Parameters

`“FMSAUC-Version”`– the name of the external function. This name must be typed exactly as it appears here, including the quotes.

`0` (zero) – a required constant.

Data type returned

Text

Description

This function returns the name and version string from the Auto Update plug-in located in the System (Windows) FileMaker Extensions (Mac OS) folder on the guest computer. If no string is found, FileMaker Pro assumes the Auto Update plug-in is missing or isn’t enabled on the guest computer.

Example

`External (“FMSAUC-Version”, 0)`

FMSAUC- FindPlugin

Format

`External (“FMSAUC-FindPlugin”, “plug-in filename”)`

Parameters

`“FMSAUC-FindPlugin”`– the name of the external function. This name must be typed exactly as it appears here, including the quotes.

`“plug-in filename”`– the name of the plug-in file. The entire parameter must be enclosed in quotes.

Data type returned

Text

Description

This function returns the version string from the plug-in version file named, which is located in the AutoUpdate folder on the FileMaker Server computer. If the plug-in version file can’t be found, the function returns `-1` (negative 1).

FileMaker Server looks for the plug-in version file in two places. First it searches for the file in an AutoUpdate folder located in the same folder that contains the hosted database. If the file is not found there, it searches for an AutoUpdate folder located in the FileMaker Server application directory.

Note Use the FileMaker Pro function Status(CurrentPlatform) to determine what operating system the guest computer is running in the event a file needs to be identified for downloading.

Example

The following example shows how to specify the external function FMSAUC-FindPlugin within a Set Field script step. Using the Set Field script step allows you to return the result of a calculation in a single field. Here the FileMaker Pro functions Status(CurrentPlatform) and Case(test1, result 1 [, test2, result 2, default result]...) are used to determine the operating system platform of the guest computer so the correct platform-specific plug-in filename is identified as the parameter. Abs(Status(CurrentPlatform)) returns 1 for Mac OS 9 and earlier or Mac OS X and 2 for Windows 95, 98, 2000, or NT. For more information about script steps and FileMaker Pro functions, see the FileMaker Pro onscreen Help.

```
Set Field [“PluginVersion”, External (“FMSAUC-FindPlugin”,
    Case (Abs(Status(CurrentPlatform)) = 2, “FMPLUGIN.FMX”,
    Status (CurrentPlatform) = 1, “FileMaker Plugin”, “FileMaker Plugin OS X”))]
```

Note In this example, the external function queries the server for plug-in version files named FMPLUGIN.FMX.txt (Windows), FileMaker Plugin.txt (Mac OS 8.6 to 9.1), or FileMaker Plugin OS X.txt (Mac OS X) and returns the correct version string based on the platform of the guest computer. For more information about the plug-in version file, see “FMSAUC-SaveVersion” on page 8.

FMSAUC-UpdatePlugin

Format

`External (“FMSAUC-UpdatePlugin”, “file to be downloaded”)`

Parameters

`“FMSAUC-UpdatePlugin”`– the name of the external function. This name must be typed exactly as it appears here, including the quotes.

`“file to be downloaded”`– the name of the plug-in or support file that you want to download. The entire parameter must be enclosed in quotes.

Data type returned

Text

Description

This function downloads the specified plug-in or support file from the server computer to the guest computer. FileMaker Server looks for the file to download in two locations: an AutoUpdate folder in the same folder as the hosted database and the AutoUpdate folder in the FileMaker Server 5.5 folder.

The file is first downloaded to a temporary folder on the guest computer. If there is a file with the same name in the System folder (Windows) FileMaker Extensions folder (Mac OS), it is moved to the Saved System folder (Windows) Saved FileMaker Extensions folder (Mac OS). If this file is a plug-in, it is disabled in FileMaker Pro before it is moved. The newly downloaded file is then copied from the temporary folder to the System folder (Windows) FileMaker Extensions folder (Mac OS) and enabled, if the downloaded file is a plug-in.

If the file successfully downloads, the function returns 0 (zero). Otherwise, the function returns an error code listed in the following table.

Error code	Description
-1	The file to be downloaded is missing from the temporary folder
-2	The Saved System folder (Windows) or Saved FileMaker Extensions (Mac OS) to contain the backup of the outdated plug-in or support file couldn’t be created on the guest computer

Error code	Description
-3	The file to be replaced on the guest computer couldn’t be deleted from the System folder (Windows) or FileMaker Extensions folder (Mac OS)
- 4	The file to be replaced couldn’t be moved to the Saved System folder (Windows) Saved FileMaker Extensions folder (Mac OS)
-5	The downloaded file can’t be copied to the System folder (Windows) or FileMaker Extensions (Mac OS)
-6	The download file must be a plug-in file
3	The Auto Update plug-in is disabled in the FileMaker Server Properties (Windows) Preferences (Mac OS)
5	The download file can’t be found in the AutoUpdate folder on the FileMaker Server computer
6	An error occurred on the computer running FileMaker Server as the file was being downloaded
100	The external function definition for FMSAUC-UpdatePlugin contains an invalid or empty parameter
101	The function call from the guest computer to the computer running FileMaker Server failed. The server computer might be running a version of FileMaker Server previous to 5.5.

If an error occurs during the downloading process, the FMSAUC-UpdatePlugin function attempts to restore all files to the state they were in when the function call executed. If an existing plug-in was disabled and moved to the Saved System folder (Windows) Saved FileMaker Extensions folder (Mac OS), it is moved back to the System folder (Windows) FileMaker Extensions folder (Mac OS) and re-enabled in FileMaker Pro on the guest computer. If the downloaded file exists in the temporary folder, it is deleted.

Important Mac OS: If the guest computer is running the Mac OS, the FMSAUC-UpdatePlugin function automatically downloads and decodes the plug-in file from MacBinary format before moving to into the FileMaker Extensions folder.

Example

The following example shows how to specify the external function FMSAUC-UpdatePlugin within a Set Field script step. Using the Set Field script step allows you to return the result of a calculation in a single field. In this case, the FileMaker Pro function Status(CurrentPlatform) is used to determine the operating system platform of the guest computer so the correct, platform-specific plug-in is downloaded by FileMaker Server (`CLNTLOG.FMX` for a Windows guest computer, `Client Log OS X` for a Mac OS X computer, or `Client Log` for a Mac OS 8.6 to 9.1 guest computer).

```
Set Field [“UpdateResult,” External (“FMSAUC-UpdatePlugin”,
    Case (Abs(Status(CurrentPlatform)) = 2, “CLNTLOG.FMX”,
    Status (CurrentPlatform) = -1, “Client Log OS X”, “Client Log”)”]
```

FMSAUC-SaveVersion

Format

`External(“FMSAUC-SaveVersion”, “version string | plug-in name”)`

Parameters

`“FMSAUC-SaveVersion”`– the name of the external function. This name must be typed exactly as it appears here, including the quotes.

`“version string | plug-in name”` – the version string from the plug-in required by the database and the name of the plug-in as a relative file path (path from the parent folder containing the FileMaker Pro file), separated by a pipe (|) character. When specifying the relative path, use colons (:) for separators. The entire parameter must be enclosed in quotes.

Note The version string can be derived by calling the external function YourPlugin-Version or hard-coded as the parameter. For more information, see the examples below.

Data type returned

Text

Description

This function saves the version string in a plug-in version file located in the FileMaker Pro folder. The filename is the name of the plug-in followed by a .TXT filename extension. If the plug-in version file is created, the function returns 0 (zero). Otherwise, the function returns 1 (one).

Examples

Windows, Mac OS 8.6 to 9.1: The following examples show two methods for saving plug-in version information in a text file.

- The following function call saves the version string `CLNTLOG 5.0` in a text file named `CLNTLOG.FMX.txt` in the FileMaker Pro 5.5 folder. In this example, `CLNTLOG.FMX` is a FileMaker Pro plug-in in Windows format.

  ```
  External (“FMSAUC-SaveVersion”, “CLNTLOG 5.0|CLNTLOG.FMX”)
  ```

- The following function call uses the version string obtained from the CLNTLOG-Version function and saves it in a text file named `Client Log.txt` in the FileMaker Pro 5.5 folder.

  ```
  External (“FMSAUC-SaveVersion”, External (“CLNTLOG-Version”, 0) & “|” &
  “Client Log”)
  ```

Mac OS X: The following function call uses the version string obtained from the CLNTLOG-Version function and saves it in a text file named `Client Log.txt` in the FileMaker Pro 5.5 folder.

```
External (“FMSAUC-SaveVersion”, External (“CLNTLOG-Version”,0) & “|” &
“::::Client Log”)
```

FMSAUC-SaveAsMacBin (Mac OS)

Format

```
External (“FMSAUC-SaveAsMacBin”, “:FileMaker Extensions:Mac OS plug-in name”)
```

Parameters

`“FMSAUC-SaveAsMacBin”`– the name of the external function. This name must be typed exactly as it appears here, including the quotes.

`“:FileMaker Extensions:Mac OS plug-in name”`– the relative file path (path from the parent folder containing the FileMaker Pro file) of the plug-in or support file that you want to encode in MacBinary format. The entire parameter must be enclosed in quotes.

Data type returned

Text

Description

This function makes a copy of the specified file, saves it in MacBinary format, appends the .bin extension to the filename, and saves it in the FileMaker Extensions folder.

Mac OS files are composed of a data fork and a resource fork. When you move a Mac OS file to a non-Mac OS system (for example, to store a plug-in on a Windows or Linux server computer), you lose its resource fork. In order to have the format of plug-in files remain consistent no matter what platform they are stored on, you can use the FMSAUC-SaveAsMacBin function to save the plug-in in MacBinary format for download to Mac OS guest computers.

Database developers should give a copy of the MacBinary file to the FileMaker Server administrator for storage in the AutoUpdate folder on the server computer so it is available for downloading, should a Mac-OS specific version of the download file be needed by a hosted database.

If the relative path is correct, the function returns 0 (zero). Otherwise, the function returns an error code such as "-43 File not found."

Example

The following example saves a copy of the "Client Log" plug-in in MacBinary format. The resulting file is named "Client Log.bin" and is stored in the FileMaker Extensions folder.

Windows and Mac OS 8.6 to 9.1:

```
External (“FMSAUC-SaveAsMacBin”, “:FileMaker Extensions:Client Log”)
```

Mac OS X:

External(“FMSAUC-SaveAsMacBin”, “::::FileMaker Extensions:Client Log”)