

FileMaker® Server 8

Guide to Updating Plug-ins



© 2001-2005 FileMaker, Inc. All Rights Reserved.

FileMaker, Inc.
5201 Patrick Henry Drive
Santa Clara, California 95054

FileMaker is a trademark of FileMaker, Inc., registered in the U.S. and other countries, and ScriptMaker and the file folder logo are trademarks of FileMaker, Inc.

FileMaker documentation is copyrighted. You are not authorized to make additional copies or distribute this documentation without written permission from FileMaker. You may use this documentation solely with a valid licensed copy of FileMaker software.

All persons and companies listed in the examples are purely fictitious and any resemblance to existing persons and companies is purely coincidental.

Credits are listed in the Acknowledgements document provided with this software.

For more information, visit our web site at www.filemaker.com.

Edition: 01

Updating plug-ins

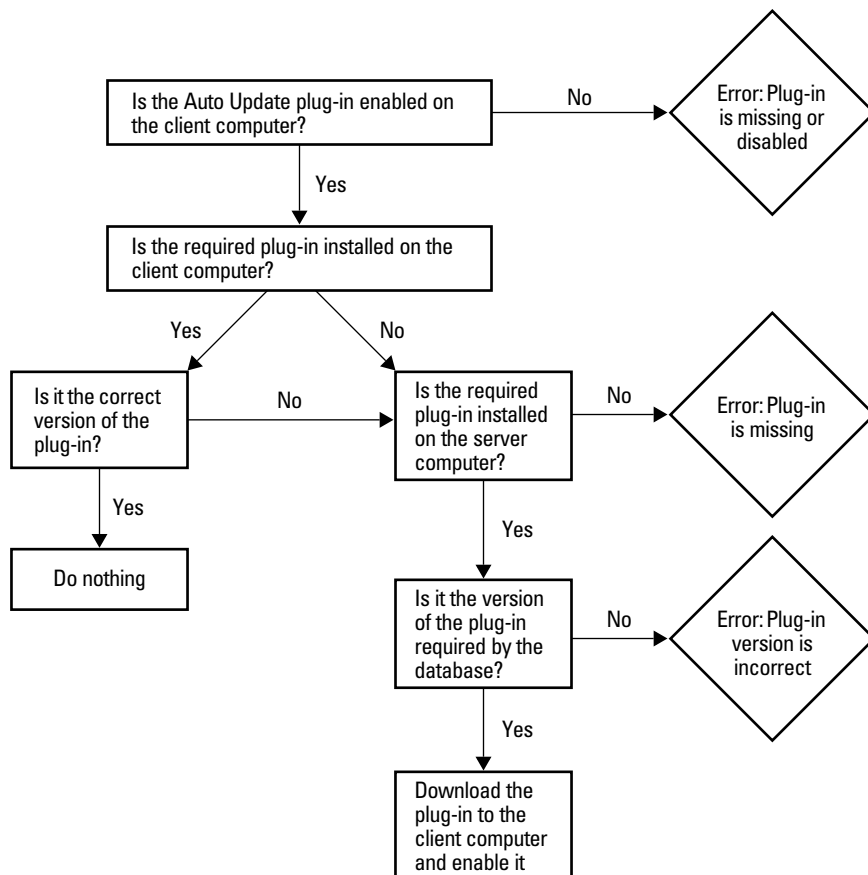
This guide describes how to use the Auto Update feature in FileMaker® Server. Auto Update ensures that FileMaker Pro database clients have the most current plug-in software installed on their computers. You can download plug-ins from FileMaker Server by including Auto Update functions in user defined scripts saved with FileMaker Pro database files. The following describes the Auto Update feature, an example script, and the Auto Update functions.

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

For information about creating plug-ins, see “Developing third-party plug-ins” in the *FileMaker Pro Advanced Development Guide*. The FileMaker Pro Advanced CD includes an example plug-in project that you can modify to include your own external functions.

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

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



For automatic update to work properly, you must:

- Install and enable the Auto Update plug-in on each FileMaker Pro client. Set up your database to call the external functions provided by the Auto Update plug-in. These functions verify the existence and version of required plug-ins on both the client and server computers and download plug-ins, if needed.
- Give plug-ins to the FileMaker Server administrator, so they can be placed in the AutoUpdate folder 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 Admin Client Connections Assistant, FileMaker Server Properties (Windows) or **Configure > Clients** (Mac OS).
- On Mac OS, plug-in files must be converted to the .tar format. FileMaker Server will automatically convert plug-ins to Mac OS clients during download, or you can manually convert plug-in versions to the .tar file format using the Mac OS tar utility. For more information, see the next section.

Preparing Mac OS plug-in files for conversion to .tar format

Before being converted to the .tar format by FileMaker Server, Mac OS plug-in files with resource forks must be processed by software that will preserve the resource fork information.

To prepare Mac OS plug-in files for conversion to .tar format:

1. Install the StuffIt Expander utility on any client computer where plug-ins will be downloaded.
StuffIt Expander is available from www.allume.com.
2. Create an archive for the plug-in using StuffIt. The sitx archive should be re-named `<filename>.fmplugin.tar`
3. Place the plug-in archive on the server computer in the appropriate folder. See “Where to store plug-ins on FileMaker Server” on page 7.

Note StuffIt must be installed on the client computer in order for the plug-in archive to be expanded.

How automatic downloading works

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

- The client is opening your database for the first time, and the plug-in that the database requires doesn't exist on the client computer.
- The client has opened your database 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 client computer.

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

The client starts FileMaker Pro and attempts to open your database hosted by FileMaker Server. 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, returning the name and version of the Auto Update plug-in available on the FileMaker Pro client computer. If plug-in information isn't returned, it is assumed that the plug-in is missing from or disabled on the client computer.
2. The external function `YourPlugIn_Version` runs, returning the name and version of the client computer plug-in that will be used in the database. If plug-in version information is not returned, it is assumed that the plug-in is missing or disabled on the client computer.
3. The external function `FMSAUC_FindPlugIn` runs, searching the server AutoUpdate folder and the server default database folder for the plug-in versions. The function returns a string, delimited by spaces, that lists all available plug-in versions.
4. The string is searched for the version returned by the `YourPlugIn_Version` function. If no version information is returned by `YourPlugIn_Version`, it is probably because the version on the server is greater than the version of the plug-in on the client computer.
5. The external function `FMSAUC_UpdatePlugIn` runs, placing the required plug-in in the FileMaker Pro Extensions folder if the option to return the latest plug-in version is enabled.

Notes

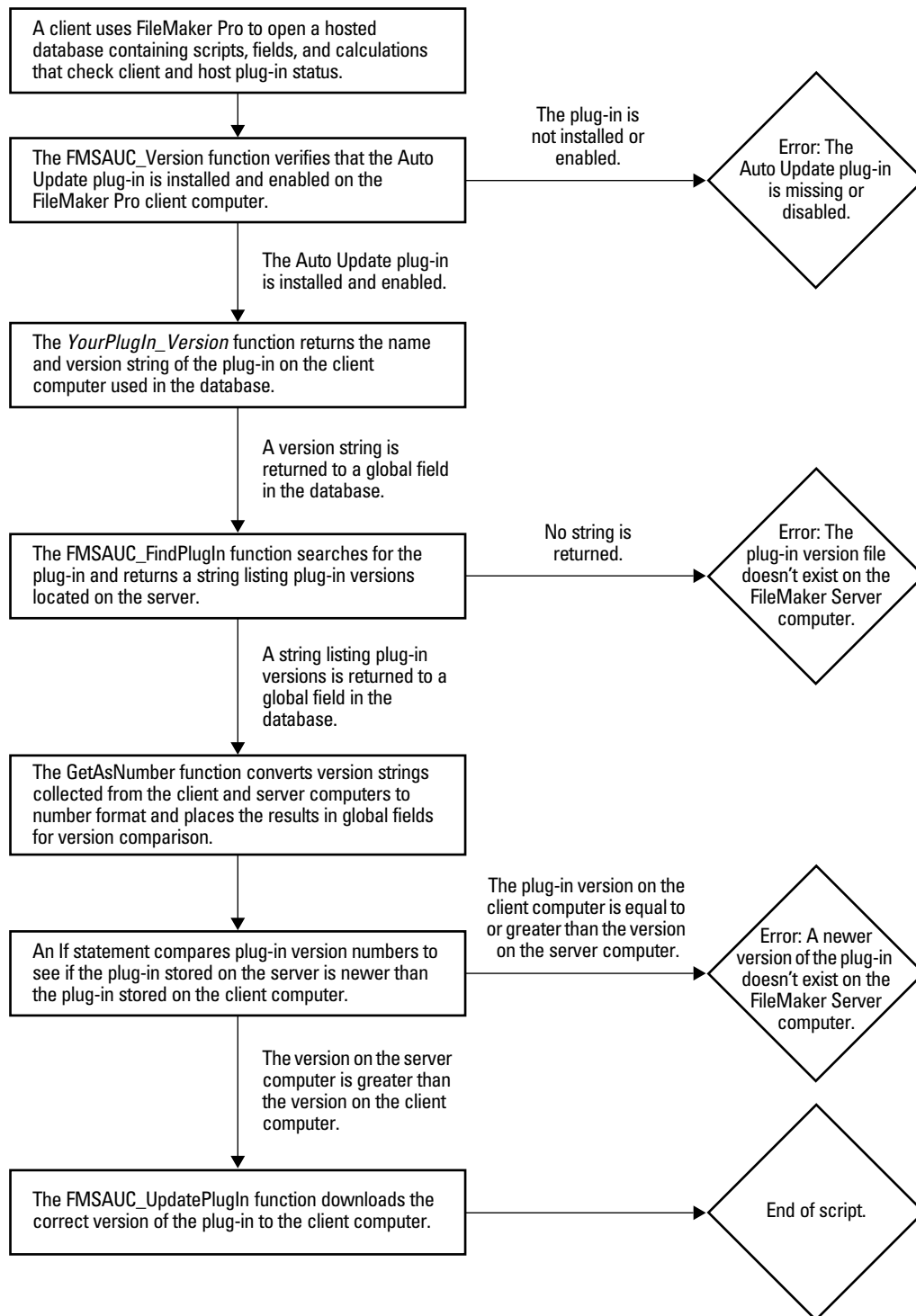
- Ask for plug-in by name and version number, separated by a space. Because spaces are used as function delimiters, plug-in names and versions cannot contain any embedded spaces.
- Do not attempt to use the Auto Update function to update itself, or the environment may become unstable and crash.

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

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

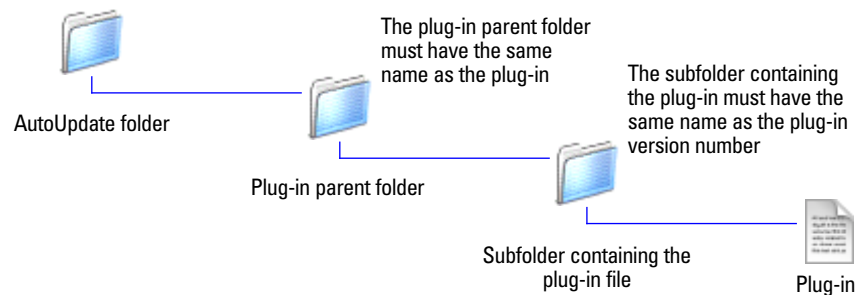
1. The external function `FMSAUC_Version` runs, returning the name and version of the Auto Update plug-in available on the FileMaker Pro client computer. If plug-in information isn't returned, it is assumed that the plug-in is missing from or disabled on the client computer.
2. The external function `YourPlugIn_Version` runs, returning the name and version of the client computer plug-in that will be used in the database. If plug-in version information is not returned, it is assumed that the plug-in is missing or disabled on the client computer.
3. The external function `FMSAUC_FindPlugIn` runs, searching the server AutoUpdate folder and the server default database folder for the plug-in versions, and returning a string that lists all available plug-in versions.
4. The string is searched for the version returned by the `YourPlugIn_Version` function. In this case, the server plug-in version is newer than the client version, so the version on the server is greater than the version of the plug-in on the client computer.
5. The external function `FMSAUC_UpdatePlugIn` runs, placing the required plug-in in the FileMaker Pro Extensions folder. The out-of-date plug-in is moved to the directory FileMaker Pro 8\Extensions\Saved\.

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



Where to store plug-ins on FileMaker Server

You must store plug-ins in the following folders on each platform:



Overview of the plug-in folder structure

Windows

To store plug-in files on a Windows server, create a folder in the AutoUpdate folder named after the plug-in, and a subfolder for each version of the plug-in. Store the plug-in file in the version subfolder.

Examples:

```
C:\Program Files\FileMaker\FileMaker Server\Data\Databases\AutoUpdate\FMS_Sample_PlugIn\1.0\
```

```
C:\Program Files\FileMaker\FileMaker Server\Data\Databases\AutoUpdate\FMS_Sample_PlugIn\1.0\FMS_Sample_PlugIn.fmx
```

```
C:\Program Files\FileMaker\FileMaker Server\Data\Databases\AutoUpdate\FMS_Sample_PlugIn\1.0\FMS_Sample_PlugIn.fmpugin.tar
```

```
C:\Program Files\FileMaker\FileMaker Server\Data\Databases\AutoUpdate\FMS_Sample_PlugIn\2.0\FMS_Sample_PlugIn.fmx
```

```
C:\Program Files\FileMaker\FileMaker Server\Data\Databases\AutoUpdate\FMS_Sample_PlugIn\2.0\FMS_Sample_PlugIn.fmpugin.tar
```

Note When Mac OS plug-ins are stored on a Windows server for use with Mac OS clients, they must be stored in the .tar archive format. For more information about archiving files using this format, launch the Mac OS Terminal application and type `man tar` at the prompt, and see “Preparing Mac OS plug-in files for conversion to .tar format” on page 4.

Mac OS

To store plug-in files on a Mac OS server, create a folder in the AutoUpdate folder named after the plug-in, and a subfolder for each version of the plug-in. Store the plug-in file in the version subfolder.

Examples:

```
/Library/FileMaker Server/Data/Databases/AutoUpdate/FMS_Sample_PlugIn
/1.0/
```

```
/Library/FileMaker Server/Data/Databases/AutoUpdate/FMS_Sample_PlugIn
/1.0/FMS_Sample_PlugIn.fmx
```

```
/Library/FileMaker Server/Data/Databases/AutoUpdate/FMS_Sample_PlugIn
/1.0/FMS_Sample_PlugIn.fmpugin.tar
```

```
/Library/FileMaker Server/Data/Databases/AutoUpdate/FMS_Sample_PlugIn
/2.0/FMS_Sample_PlugIn.fmx
```

```
/Library/FileMaker Server/Data/Databases/AutoUpdate/FMS_Sample_PlugIn
/2.0/FMS_Sample_PlugIn.fmpugin.tar
```

Important To function properly on Mac OS, plug-ins and the folders in which they are stored must have the following file permissions:

| File or folder | Group | Permissions |
|------------------------------|----------|---------------------------|
| Plug-in parent folder | fmsadmin | Read and execute by group |
| Subfolder containing plug-in | fmsadmin | Read and execute by group |
| Plug-in file | fmsadmin | Read and execute by group |

To do this:

1. Launch the Terminal application ([hard disk]/Applications/Utilities/Terminal).
2. Navigate to the parent folder of the file or folder whose permissions you intend to change and enter the following at the command line:

```
chmod g+rx <filename or folder>
```

or

```
chmod g+wrx <filename or folder>
```

The `g+rx` is necessary because scripts and plug-ins must have the group read and executable bits enabled. Use the `g+wrx` form to permit write permission as well. Some plug-ins or scripts that use preferences or folders of additional files may require write permission to those files or folders.

Setting up Auto Update in your database

There are several ways to set up Auto Update in your database. This guide describes one way, which uses simple script steps and global fields, and is based on the sample file named AutoUpdatePlugin.fp7 installed with FileMaker Server. This sample file is also available on the FileMaker, Inc. web site.

Overview

To set up plug-in version checking in your database, you need to write a few simple script steps that run when a client 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 client 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.

To set up plug-in version checking in your database

1. Open your database using FileMaker Pro and enable the Auto Update plug-in in the Preferences dialog box.

Note Remind the server administrator to turn on Auto Update in FileMaker Server via the Client Connections Assistant, FileMaker Server Properties dialog box (Windows), or **Configure > Clients** (Mac OS).

2. Create a layout to contain:
 - fields defined with the global storage option to collect plug-in version information or result codes
 - a button to manually run the script that creates a plug-in version information file for storage on the server

For an example layout that contains all the fields and the button you need, see the STARTUP layout in the AutoUpdatePlugin.fp7 sample file.

3. Write a script that uses the `FMSAUC_Version` function, which returns the name and version of the Auto Update plug-in available in FileMaker Pro. If the name and version string is not returned, FileMaker Pro assumes the Auto Update plug-in is missing or isn't enabled on the client computer.
4. Write a script that uses the `YourPlugIn_Version` function to check the version of the plug-in on the client 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.fp7 sample file. Information about the `YourPlugIn_Version` function should be included in the third-party plug-in documentation.

5. 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 in the global field named `Remote_Version` in your Auto Update layout.

For script syntax, see the Remote Plug-in Check script in the AutoUpdatePlugin.fp7 sample file. For information about the `FMSAUC_FindPlugIn` function, see "FMSAUC_FindPlugIn" on page 11.

6. 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 client and server computers.

For script syntax, see the Get Version Numbers script in the AutoUpdatePlugin.fp7 sample file. For information about the `GetAsNumber` function, see the FileMaker Pro onscreen Help.

7. Write a script to download the plug-in from the server computer, if the version on the client computer doesn't exist or is outdated.

For script syntax, see the Download Plug-In script in the AutoUpdatePlugin.fp7 sample file. For information about the FMSAUC_UpdatePlugIn function, see “FMSAUC_UpdatePlugIn” on page 11.

8. Write a “parent” script to:

- make sure the Auto Update plug-in exists and is enabled on the client 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.fp7 sample file. For information about the FMSAUC_Version function, see “FMSAUC_Version” on this page.

External functions

Auto Update consists of the following external functions.

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

| This function | Does this |
|---------------------|--|
| FMSAUC_Version | Returns the name and version of the Auto Update plug-in that exists in the Extensions folder on the client computer. If no version is returned, FileMaker Pro assumes that the plug-in is missing or isn't enabled on the client computer. |
| FMSAUC_FindPlugIn | Returns a string listing the plug-in versions located in the AutoUpdate folder or the default database 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 AutoUpdate folder or the default database folder on the server computer or returns an error code if the file can't be downloaded. |

FMSAUC_Version

| | |
|---------------------------|--|
| Format | FMSAUC_Version() |
| Parameters | Constant integer; 0 |
| Data type returned | Text |
| Description | This function returns the name and version string from the Auto Update plug-in located in the FileMaker Pro Extensions folder on the client computer. If no string is found, FileMaker Pro assumes the Auto Update plug-in is missing or isn't enabled on the client computer. |
| Example | FMSAUC_Version(0) |

FMSAUC_FindPlugIn

| | |
|---------------------------|---|
| Format | FMSAUC_FindPlugIn("plug-in-name") |
| Parameters | plug-in-name – the name of the plug-in file. The parameter must be enclosed in quotes. |
| Data type returned | Text |
| Description | <p>This function returns a string listing the plug-in versions located in the FileMaker Server AutoUpdate folder or the default database folder. If the plug-in can't be found, the function returns -1 (negative 1).</p> <p>FileMaker Server looks for the plug-in in two places. First, it searches the 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 default database folder.</p> |
| Example | <p>The following example shows how to specify the external function FMSAUC_FindPlugIn within a Set Field script step. The Set Field script step allows you to return the result of a calculation in a single field. For more information about script steps and FileMaker Pro functions, see the FileMaker Pro onscreen Help.</p> <pre>Set Field [dbname::Remote_Version; FMSAUC_FindPlugIn("SamplePlugIn")]</pre> <p>This example returns versions in the following format:</p> <p>1.0 1.2 1.5 2.0 3.3...</p> |

FMSAUC_UpdatePlugIn

| | |
|---------------------------|--|
| Format | FMSAUC_UpdatePlugIn("plug-in-name version") |
| Parameters | plug-in-name version – the name of the plug-in or support file that you want to download and the plug-in version number. The parameter must be enclosed in quotes. |
| | Note There must be a space between the plug-in name and the version number. |
| Data type returned | Text |
| Description | <p>This function downloads the specified plug-in or support file from the server computer to the client computer. FileMaker Server looks for the plug-in in two places. First, it searches the 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 default database folder.</p> |
| Example | <pre>FMS_UpdatePlugIn("SamplePlugIn 1.5").</pre> <p>If there is a file with the same name in the client's Extensions folder, it is moved to the Extensions\Saved folder. If this file is a plug-in, it is disabled in FileMaker Pro before it is moved. The newly downloaded file is then copied to the Extensions folder and enabled, if the downloaded file is a plug-in.</p> <p>If the file successfully downloads, the function returns 0 (zero). Otherwise, the function returns an error code listed in the following table.</p> |

| Error code | Description |
|-------------------|---|
| -1 | The file to be downloaded is missing from the temporary folder |
| -2 | The Extensions\Saved folder to contain the backup of the outdated plug-in or support file couldn't be created on the client computer |
| -3 | The file to be replaced on the client computer couldn't be deleted from the Extensions folder |
| -4 | The file to be replaced couldn't be moved to the Extensions\Saved folder |
| -5 | The downloaded file can't be copied to the Extensions folder |
| -6 | The download file must be a plug-in file |
| 3 | The Auto Update plug-in is disabled in the FileMaker Server Admin Client Connections Assistant, FileMaker Server Properties (Windows) or Configure > Clients (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 client computer to the computer running FileMaker Server failed. The server computer might be running a previous version of FileMaker Server. |

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 Extensions\Saved folder, it is moved back to the Extensions folder and re-enabled in FileMaker Pro on the client computer.