

Bootstrapping MBS FileMaker Plugin

How to automatically install MBS Plugin in FileMaker without having the plugin?

We have gzip versions of our plugins here: [gzip downloads](#). There you find the current URLs for all our plugins and the [version.json](#) with the version information. You may do a script to automatically install the plugins.

Now we got the question how to load the MBS Plugin with Insert From URL for an installation, so here is an example script:

normal release here

```
Set Variable [ $URL ; Value: "https://www.monkeybreadsoftware.com/filemaker/files/gzip/version.json" ]
```

```
Set Variable [ $JSON ; Value: "" ]
```

```
Insert from URL [ Select ; With dialog: Off ; Target: $JSON ; $URL ; cURL options: "--FM-text-encoding utf-8 --location --max-time 10 " ; Do not automatically encode URL ]
```

```
If [ Get(LastError) ≠ 0 ]
```

```
    Show Custom Dialog [ "Failed to query MBS server." ;
```

```
    Get(LastErrorDetail) ]
```

```
    Exit Script [ Text Result: "Failed to query server." ]
```

```
End If
```

#

```
Set Variable [ $date ; Value: JSONGetElement ( $json ; "date" ) ]
```

```
Set Variable [ $versionNumber ; Value: JSONGetElement ( $json ; "versionNumber" ) ]
```

```
Set Variable [ $versionText ; Value: JSONGetElement ( $json ; "versionText" ) ]
```

#

```
Set Variable [ $URLMac ; Value: JSONGetElement ( $json ; "download.mac" ) ]
```

```
Set Variable [ $URLWin32 ; Value: JSONGetElement ( $json ; "download.win32" ) ]
```

```
Set Variable [ $URLWin64 ; Value: JSONGetElement ( $json ; "download.win64" ) ]
```

```
Set Variable [ $URLLinux ; Value: JSONGetElement ( $json ; "download.linux" ) ]
```

#

```
Set Variable [ $LoadedCount ; Value: 0 ]
```

Load Windows 32-bit

```
Set Variable [ $Result ; Value: "" ]
```

```
Insert from URL [ Select ; With dialog: Off ; Target: $result ; $URLWin32 ; cURL options: "--FM-return-container-variable --location " ; Do not automatically encode URL ]
```

```
If [ Get(LastError) = 0 and Length($Result) > 1000000 ]
```

```
    Set Field [ Install Plugin Update if needed::Plugin File Win 32bit ; $result ]
```

```
    Commit Records/Requests [ With dialog: Off ]
```

```

        Set Variable [ $LoadedCount ; Value: $LoadedCount + 1 ]
End If
#
# Load Windows 64-bit
Set Variable [ $Result ; Value: "" ]
Insert from URL [ Select ; With dialog: Off ; Target: $Result ; $URLWin64 ; cURL
options: "--FM-return-container-variable --location " ; Do not automatically encode
URL ]
If [ Get(LastError) = 0 and Length($Result) > 1000000 ]
    Set Field [ Install Plugin Update if needed::Plugin File Win 64bit ; $result ]
    Commit Records/Requests [ With dialog: Off ]
    Set Variable [ $LoadedCount ; Value: $LoadedCount + 1 ]
End If
#
# Load macOS
Set Variable [ $Result ; Value: "" ]
Insert from URL [ Select ; With dialog: Off ; Target: $Result ; $URLMac ; cURL
options: "--FM-return-container-variable --location " ; Do not automatically encode
URL ]
If [ Get(LastError) = 0 and Length($Result) > 1000000 ]
    Set Field [ Install Plugin Update if needed::Plugin File Mac ; $result ]
    Commit Records/Requests [ With dialog: Off ]
    Set Variable [ $LoadedCount ; Value: $LoadedCount + 1 ]
End If
#
# Load Linux
Set Variable [ $Result ; Value: "" ]
Insert from URL [ Select ; With dialog: Off ; Target: $Result ; $URLLinux ; cURL
options: "--FM-return-container-variable --location " ; Do not automatically encode
URL ]
If [ Get(LastError) = 0 and Length($Result) > 1000000 ]
    Set Field [ Install Plugin Update if needed::Plugin File Linux ; $result ]
    Commit Records/Requests [ With dialog: Off ]
    Set Variable [ $LoadedCount ; Value: $LoadedCount + 1 ]
End If
#
# all done
If [ $LoadedCount > 0 ]
    Set Field [ Install Plugin Update if needed::Version Display ; $versionText ]
    Set Field [ Install Plugin Update if needed::Version Number ;
$versionNumber ]
    Commit Records/Requests [ With dialog: Off ]
End If
Show Custom Dialog [ "Plugins loaded" ; "Loaded " & $LoadedCount & " of 4
plugin files." ]

```

The script starts with loading out version.json file to get the JSON, which currently looks like this:

```
{
  "date": "2022-09-06",
  "versionNumber": "12040006",
  "versionText": "12.4.0.06",
  "download": {
    "mac": "https://www.monkeybreadsoftware.com/filemaker/files/gzip/
MBS.fmplugin.gz",
    "linux": "https://www.monkeybreadsoftware.com/filemaker/files/gzip/
MBS.fmx.gz",
    "win32": "https://www.monkeybreadsoftware.com/filemaker/files/gzip/
MBS.fmx32.gz",
    "win64": "https://www.monkeybreadsoftware.com/filemaker/files/gzip/
MBS.fmx64.gz"
  }
}
```

We extract all the values into local variables. Then we try to load each URL into a variable. If that succeeds and the downloaded container value has more than one million bytes, we store it in the container field. In case something went wrong and the file is missing and we get an error page or redirect to the file listing, the error is either not zero or the length is much smaller.

When all are done, we store the new version number, so you can use the Install script. Feel free to adjust the script to maybe add more automation or more error checks, e.g. for JSON. As you see we currently try all 4 downloads and then report how many worked.

The example script will be included with next plugin version in the "Install Plugin Update if needed.fmp12" database file.

Please let us know if you have questions for this.

see also

Plugin Installation and Registration

- [Automatically install MBS FileMaker Plugin \(PDF\)](#)
- [Install MBS Plugin on a FileMaker Server \(PDF\)](#)
- [Why MBS Plugin fails to load \(PDF\)](#)
- [FileMaker Server Plugin Installation \(PDF\)](#)
- [Installation of MBS Plugin on FileMaker Server for Windows \(PDF\)](#)
- [FileMaker Server Security and plugins \(PDF\)](#)
- [MBS FileMaker Plugin registration dialogs \(PDF\)](#)
- [Installing MBS Plugin on FileMaker Server \(PDF\)](#)
- [Properly register MBS FileMaker Plugin \(PDF\)](#)
- [Please don't deactivate plugins](#)
- [The MBS Script Step \(PDF\)](#)
- [Getting Started \(PDF\)](#)