M-FILES CORPORATION

INSTALLING M-FILES DESKTOP, M-FILES SERVER, AND M-FILES ADMIN WITH ADVANCED OPTIONS

CUSTOMIZATION, COMMAND-LINE OPTIONS, AND CENTRALIZED DEPLOYMENT

LAST UPDATED 11 JUNE 2025

VERSION 2.6

CONTENTS

1. CHANGE HISTORY

Version	Description
1.0	Initial published version.
1.1	Section 8 added.
1.2	Section 5.7 modified.
1.3	Added descriptions for the XML elements used for defining the connections in sections 3.2.1 and 3.2.2.
1.4	Removed the sections *M-Files Drive Letter* (was section 3.2.4) and *Removing Registry Entries* (was section 3.2.7).
1.5	Notes about signing the Installation Package when using a certificate added in section 3.1.
1.6	HTTP proxy setting added in sections 3.2.1 and 3.2.2.
1.7	Spelling mistake corrected in section 3.1. Cross-references updated.
1.8	Added a note to section 5.7.
1.9	Added information about overriding the default functionality in installation in section 5.6.
2.0	Added information about new options in the XML file (section 3.2).
2.1	Minor correction to the example in section 3.2.2.
2.2	gRPC added as a connection protocol to sections 3.2.1 and 3.2.2.
2.3	Changes related to the customization script.
2.4	Document name changed.
2.5	Section 3.2.7 updated.
2.6	Removed references to 32-bit installers and some unnecessary version information. Updated section 8 to refer to Advanced Vault Settings.

2. OVERVIEW

The M-Files installer is a Windows Installer package (MSI file). Most users install M-Files by simply double-clicking the MSI file and following the instructions of the setup program. For basic installation instructions in full UI mode, see the M-Files user guide.

IT administrators may wish to customize the behavior of the M-Files setup program or deploy M-Files in a centralized, automated fashion. This document describes the available customization possibilities and command-line options and includes a description of how M-Files Desktop can be automatically deployed to workstations via a Group Policy object (GPO) in Windows.

2.1. M-FILES INSTALLATION PACKAGE VARIANTS

The M-Files installation package is available in three variants:

- Full installer
- "Client only" installer
- "Client and server tools only" installer

The full installer includes all components of M-Files: M-Files Desktop, M-Files Server, and M-Files Server Tools (M-Files Admin).

The "client only" and "client and server tools only" installers can be used for simplifying the deployment of the client components of M-Files to users' computers. Compared to the full installer, the MSI file is smaller, and there are fewer options when the setup program is run.

2.2. CUSTOMIZING THE INSTALLATION PACKAGE

You can customize the installation package of M-Files by running the customization script attached to the same multi-file document as this guide. Especially the ability to customize the "client only" installer to automatically install the needed vault connections helps simplify the distribution of M-Files Desktop to users.

Section 3 of this document describes how the installer can be customized to include pre-configured vault connections for M-Files Desktop and/or pre-configured server connections for M-Files Server Tools (M-Files Admin). Additionally, various other settings can be controlled by the customization mechanism.

2.3. COMMAND-LINE OPTIONS

The behavior of the M-Files setup program can be controlled with command-line options.

Standard Windows Installer command-line options such as /quiet and /log are supported. Additionally, various M-Files-specific properties can be specified from the command line to modify the behavior of the installation.

See section 5 for information on the command-line options.

2.4. CENTRALIZED DEPLOYMENT VIA GROUP POLICY

You can use the Group Policy feature of Windows to automatically distribute M-Files to client computers. Alternatively, you may use any other centralized deployment mechanism that you are familiar with.

Section 9 provides instructions on how to create a Group Policy object (GPO) that automatically installs M-Files Desktop to the specified client computers.

3. CUSTOMIZING THE INSTALLATION PACKAGE

You can customize the installation package of M-Files by running the customization script attached to the same multi-file document as this guide. The script modifies the MSI file.

Note: These instructions apply to the 2.3 version of the script file. The 1.0 version is included in the multi-file document as well because of backwards compatibility.

The vault connections and server connections are specified in a simple XML file. The customization script reads the XML file and applies the necessary changes to the installation package.

3.1. COMMAND LINE SYNTAX

Use the following command line syntax to customize the installation package:

```
cscript <CustomizeInstaller filename> xml <source MSI file> <XML file> <target MSI file> <silent> <sign>
```

`CustomizeInstaller filename`	For example **`CustomizeInstaller1.0.vbs`** or **`CustomizeInstaller2.3.vbs`**.
`xml`	The word "xml" specifies the operating mode for the script. Must always be specified.
`<source MSI file>`	The M-Files installation package to use as the basis. Typically the client-only default installer.
`<XML file>`	The XML file that specifies the desired customizations, such as vault connections and/or server connections.
`<target MSI file>`	The new MSI file to create.
`<silent>`	If True, prints no output. If False, prints information on the customizations.
`<sign>`	If True, the target MSI file is digitally signed by using the M-Files Corporation certificate. Requires that the certificate is installed on the computer. If False, the target MSI file is not digitally signed.

For example, the following command could be used to create a customized version of the M-Files 11.0.4210.0 client-only installer (line breaks have been added for clarity):

```
cscript CustomizeInstaller2.3.vbs
   xml
   "C:\Temp\M-Files_x64_eng_client_11_0_4210_0.msi"
   "C:\Temp\MyConnections.xml"
   "C:\Temp\M-Files_x64_eng_client_11_0_4210_0_MyCompany.msi"
   False
   False
```

Note: Partners and customers cannot sign the installer package with the M-Files Corporation certificate. Typically, there is no need to sign the customized MSI packaged at all. However, if a reseller or customer wants to sign the installation package, they need to acquire their own signing certificate and modify the "CustomizeInstaller" VBS script so that it uses that certificate.

3.2. XML FILE SCHEMA

The attached file **`CustomizeInstaller.xml`** can be used as a template for creating the actual XML file.

3.2.1 VAULT CONNECTIONS FOR M-FILES DESKTOP

The XML file may specify zero, one or more vault connections that are automatically configured during the installation of the M-Files Desktop component. The XML file can specify any of the values that the vault connection details in the Windows registry contain. When you specify one or more vault connections to be configured during the installation, the XML schema must contain a `VaultConnections` element with the value `True` under the root level (see the example further below).

This table describes the XML elements used for defining the connection:

Element	Description	Value
ServerVaultName	The name of the vault.	For example `Vault 1`.

	Note that the name cannot contain any umlaut characters.	
ServerVaultGUID	The GUID of the vault.	For example `{13031BF9-B04A-4E9E-AABD-B11C8059084C}`.
ProtocolSequence	The protocol for the server connection.	`ncacn_ip_tcp` = TCP/IP `ncacn_http` = HTTPS `ncalrpc` = LPC `ncacn_spx` = SPX `grpc` = gRPC `grpc-local` = local gRPC
NetworkAddress	The DNS name of the server.	For example `server.company.com`.
Endpoint	Communication endpoint on which the server listens for calls.	For example 2266.
AuthType	Defines the authentication type.	`#1` = current Windows user `#2` = specific Windows user `#3` = M-Files user
AutoLogin	Defines whether the user is logged in to the vault when Windows is started.	`#0` = disabled `#1` = enabled
SPN	Service Principal Name. Should only be used if specifically required.	The value is normally not set.
MinimumAuthenticationLevel	Defines whether the server connection is encrypted.	`#1` = unencrypted connection `#6` = encrypted connection
HTTPProxy	HTTP proxy. Should only be used if specifically required. Supported in M-Files builds 18.12.XXXX.XX and later.	The value is normally not set. If needed, port is separated with colon.

EXAMPLE

```
<root>

  <VaultConnections>True</VaultConnections>

  <Client>

    <Vaults>

      <Vault Name="My Connection 1">
        <ServerVaultName>Vault 1</ServerVaultName>
        <ServerVaultGUID>{13031BF9-B04A-4E9E-AABD-B11C8059084C}</ServerVaultGUID>
        <ProtocolSequence>ncacn_ip_tcp</ProtocolSequence>
        <NetworkAddress>server.company.com</NetworkAddress>
        <Endpoint>2266</Endpoint>
        <AuthType>#1</AuthType>
        <AutoLogin>#1/AutoLogin>
        <SPN></SPN>
        <MinimumAuthenticationLevel>#1</MinimumAuthenticationLevel>
        <HTTPProxy></HTTPProxy>
      </Vault>

    </Vaults>

  </Client>

</root>
```

See the Windows registry on a computer with M-Files Desktop installed for the values that you need to specify in order to include a specific vault connection.

Key name: Software\Motive\M-Files\<version>\Client\MFClient\Vaults

Computer-specific vault connections are specified under the HKEY_LOCAL_MACHINE subtree in the registry, while user-specific vault connections are specified under the HKEY_CURRENT_USER subtree. The installation package installs all included vault connections as computer-specific connections under the HKEY_LOCAL_MACHINE subtree in the registry.

REG_SZ values are entered as such in the XML elements.

REG_DWORD values such as **AuthType**, **AutoLogin**, and **MinimumAuthenticationLevel** must be prefixed with the # sign.

3.2.2 SERVER CONNECTIONS FOR M-FILES ADMIN

The XML file may specify zero, one or more server connections that are automatically configured during the installation of the M-Files Server Tools (M-Files Admin) component. The XML file can specify any of the values that the server connection details in the Windows registry contain. When you specify one or more server connections to be configured during the installation, the XML schema must contain a `ServerConnections` element with the value `True` under the root level (see the example further below).

This table describes the XML elements used for defining the connection:

Element	Description	Value
ProtocolSequence	The protocol for the server connection.	`ncacn_ip_tcp` = TCP/IP `ncacn_http` = HTTPS `ncalrpc` = LPC `ncacn_spx` = SPX `grpc` = gRPC `grpc-local` = local gRPC
NetworkAddress	The DNS name of the server.	For example `server.company.com`.
Endpoint	Communication endpoint on which the server listens for calls.	For example 2266.
Authentication	Defines the authentication type.	#1 = current Windows user #2 = specific Windows user #3 = M-Files user
MinimumAuthenticationLevel	Defines whether the server connection is encrypted.	#1 = unencrypted connection #6 = encrypted connection
HTTPProxy	HTTP proxy. Should only be used if specifically required. Supported in M-Files builds 18.12.XXXX.XX and later.	The value is normally not set. If needed, port is separated with colon.

EXAMPLE

```
<root>

  <ServerConnections>True</ServerConnections>

  <ServerTools>

    <Servers>

      <Server Name="My Server">
        <ProtocolSequence>ncacn_ip_tcp</ProtocolSequence>
        <NetworkAddress>server.company.com</NetworkAddress>
        <Endpoint>2266</Endpoint>
        <Authentication>#1</Authentication>
        <MinimumAuthenticationLevel>#1</MinimumAuthenticationLevel>
        <HTTPProxy></HTTPProxy>
      </Server>

    </Servers>

  </ServerTools>

</root>
```

See the Windows registry on a computer with M-Files Server Tools (M-Files Admin) installed for the values that you need to specify in order to include a specific server connection.

Key name: HKEY_CURRENT_USER\Software\Motive\M-Files\<version>\ServerTools\MFAdmin\Servers

REG_SZ values are entered as such in the XML elements.

REG_DWORD values such as **Authentication** and **MinimumAuthenticationLevel** must be prefixed with the # sign.

3.2.3 AUTOMATIC UPDATES

To disable the automatic checking for software updates, add the following element to the XML file:

Element path: **root/Common/AutomaticUpdates/CheckForUpdates**
Element content: **0** (0 = disable, 1 = enable)

3.2.4 DESKTOP ICONS

To prevent M-Files Setup from installing any desktop icons, add the following element to the XML file:

Element path: **root/Setup/NoDesktopIcons**
Element content: (no content)

3.2.5 REMOVING PREVIOUS INSTALLATIONS

You can set previous M-Files installations to be removed during the installation by adding the following element to the XML file:

Element path: **root/Setup/RemoveAllPreviousVersions**
Element content: (no content)

3.2.6 ALLOWING THE INSTALLATION OF OLDER VERSIONS

You can allow updating M-Files over a specific version even if a newer version has already been installed by adding the following element to the XML file:

Element path: **root/Setup/CanInstallOlder**
Element content: (no content)

3.2.7 CUSTOM REGISTRY ENTRIES

The XML file can specify custom registry entries to be written during installation. You can use the mechanism to specify M-Files settings that are controlled by registry settings in the HKEY_LOCAL_MACHINE subtree.

> **Important**: In general, use this mechanism only to write registry values to the **SOFTWARE\Motive\M-Files\[MVERSIONSTRING]\<feature>** keys. It is technically possible to use the mechanism to write to other parts of the registry but be careful if you do that. Registry entries that the installer writes are removed during uninstallation. Even an upgrade installation can remove the entries because the previous version is uninstalled at the end of the installation of the new version. The **[MVERSIONSTRING]** placeholder will be automatically replaced by the installed version (for example, 22.8.11717.3) during installation.

For registry entries in the HKEY_CURRENT_USER subtree, you should use some other distribution mechanism, such as Group Policy object (GPO). Registry settings in the HKEY_CURRENT_USER subtree cannot be reliably applied at the time of M-Files installation because new users may log on to Windows on the computer even after the installation. Such registry settings are better controlled by a Group Policy object (GPO). See section 8 for instructions on how to distribute client-side registry settings to the users' HKEY_CURRENT_USER subtree with a Group Policy object (GPO), taking advantage of the environment variable MFilesVersion.

To specify registry settings for the HKEY_LOCAL_MACHINE subtree, add one or more **RegistryEntry** elements to the XML file. The RegistryEntry elements must be contained in a **RegistryEntries** element under one of the feature elements (**Common**, **Client**, **Server**, or **ServerTools**). The registry entries will be installed only if the containing feature is installed.

Example:

```
<root>

  <Client>

    <RegistryEntries>

      <RegistryEntry>
        <KeyName>SOFTWARE\Motive\M-Files\[MVERSIONSTRING]\Client\MFClient</KeyName>
        <ValueName>AutomaticLogoutTimeoutInMinutes</ValueName>
        <ValueData>#60</ValueData>
      </RegistryEntry>

    </RegistryEntries>

  </Client>

</root>
```

All values are written as string values (**REG_SZ**) unless you specify a prefix for the value in the ValueData element.

If the value is prefixed by **#**, the value is interpreted and stored as an integer (**REG_DWORD**).

If the value is prefixed by **#%**, the value is interpreted and stored as an expandable string (**REG_EXPAND_SZ**).

If the value is prefixed by **#x**, the value is interpreted and stored as a hexadecimal value (**REG_BINARY**).

If the value contains the sequence tilde **[~]**, then the value is interpreted as a Null-delimited list of strings (**REG_MULTI_SZ**). For example, to specify a list containing the three strings a, b and c, use "a[~]b[~]c".

Note: M-Files is sensitive to the types of the registry values. If you specify a registry value with an incorrect type (e.g., a REG_DWORD value instead of a REG_SZ value), M-Files services may not start. If you suspect such a situation, see the Windows Event Log for more information on the error.

4. EXECUTING THE CUSTOMIZED INSTALLATION PACKAGE

The customized installation package can be executed in the same way as the original installation packages, e.g., by double-clicking the MSI file to run it in UI mode, or by executing in silent mode from the command line.

4.1. NEW VS. UPGRADE INSTALLATIONS

A customized installation package is often created for the purpose of simplifying the deployment of M-Files Desktop to users' computers. In this case, the "client only" installer should be used as the basis. Users can run the customized "client only" installer with default options, and the setup program will behave as follows:

New installation (the computer does not have a previous version of M-Files installed):

The setup program will install M-Files Desktop on the user's computer, including the pre-configured vault connections.

Upgrade installation (the computer has a previous version of M-Files installed):

The setup program will install the new version of M-Files Desktop on the user's computer and at the end of the installation will remove the previous version.

All settings from the previous version will automatically be migrated to the new version. However, any pre-configured vault connections in the new version's installation package will take precedence if the previous version has vault connections with the same name. For example, if the new installation package's connection details for vault connection with the name "Our Vault" differs from the connection details of a vault connection with the same name in the previous version, the new installation package's connection details will be taken into use.

Note that performing an upgrade installation by using the "client only" installer will succeed only if the previous version of M-Files on the user's computer does not have other M-Files components installed, such as M-Files Server. If the previous version has such components installed that the "client only" installation package does not contain, then attempting to run the "client only" installer on the computer will display an error message and exit.

4.2. SILENT INSTALLATION

The installer can be run in silent mode by specifying the standard Windows Installer command-line options for silent installation. For example, the following command line will run a customized "client only" installer in silent mode, performing either a new installation or an upgrade installation, depending on if the computer has a previous version installed:

msiexec /package M-Files_x64_eng_client_11_0_4210_0_MyCompany.msi /quiet

The above command line runs a completely silent installation. Often it is better to run an unattended installation but still show a progress dialog in order to be able to see when the installer has finished running. The following command line runs the installation without prompting the user for any information and displays a progress dialog while running:

msiexec /package M-Files_x64_eng_client_11_0_4210_0_MyCompany.msi /qb!

When running the installer in silent mode from the command line, you must ensure that the installer is executed with elevated privileges. For example, you can start an elevated Command Prompt window by using the **Run as Administrator** command from the shortcut menu and then execute the msiexec command line in the elevated Command Prompt window. If you attempt to run a silent installation from a non-elevated Command Prompt window, the installation will fail silently. The Application log in Event Viewer will contain information on the error.

4.3. UPGRADE OPTIONS

When an upgrade installation is performed, that is, when there is a previous version of M-Files already installed on the computer and the installer is run with default options, all settings from the previous version will automatically be migrated to the new version.

In the default mode of the installer, any pre-configured vault connections in the new version's installation package will take precedence if the previous version has vault connections with the same name. For example, if the new installation package's connection details for vault connection with the name "Our Vault" differs from the connection details of a vault connection with the same name in the previous version, the new installation package's connection details will be taken into use.

This default behavior can be overridden by specifying the M_MIGRATIONOPTIONS_CLIENT property from the command line when executing the installer, or by using Orca or another MSI editor to edit this property in the MSI file. Supported values are:

-mergeclientvaults This is the default value in a customized installer that includes pre-configured vault connections. In this mode, the pre-configured vault connections in the installer will be merged with the existing vault connections of the previous version on the user's computer. In case of a vault connection with the same name but different connection details, the connection details specified in the new version's installation package will be used.

(empty) If the property is missing or its value is empty, the installer will perform a full migration of the previous version's vault connections. The pre-configured vault connections in the installer will be written as well, but in case of a vault connection with the same name but different connection details, the connection details used with the previous version will override the connection details specified in the new version's installation package.

-skipclientvaults In this mode, the installer will not migrate vault connections from the previous version. Only the pre-configured vault connections in the installer will be installed.

In most cases, the default "mergeclientvaults" behavior is desirable. However, if preserving the previous version's vault connection details is important in case of a name conflict, then the following command line can be used to run the installer:

```
msiexec
    /package M-Files_x64_eng_client_11_0_4210_0_MyCompany.msi
    M_MIGRATIONOPTIONS_CLIENT=""
    /qb!
```

If an upgrade installation should completely replace the vault connections, i.e., not migrate previous version's vault connections at all, the following command line can be used to run the installer:

```
msiexec
    /package M-Files_x64_eng_client_11_0_4210_0_MyCompany.msi
    M_MIGRATIONOPTIONS_CLIENT="-skipclientvaults"
    /qb!
```

For pre-configured server connections in M-Files Server Tools (M-Files Admin), the corresponding property's name is **M_MIGRATIONOPTION_SERVERTOOLS**. The supported values for the property are "**-mergeserverconnections**", empty, and "**-skipserverconnections**".

5. M-FILES SETUP COMMAND-LINE OPTIONS

The M-Files setup program is a Windows Installer MSI package. You can execute it in full UI mode by double-clicking the MSI package. Additionally, you can run M-Files Setup from the command line, optionally in silent mode.

Standard Windows Installer command-line options such as /quiet and /log can be used with M-Files Setup. Additionally, various M-Files-specific properties can be specified from the command line to modify the behavior of the installation.

This section lists typically used command-line options, whether they are standard Windows Installer options or M-Files-specific options.

5.1. GENERIC COMMAND-LINE SYNTAX

You can run M-Files Setup from the command line by invoking with Windows Installer executable **msiexec.exe**. The generic syntax for installing M-Files from the command line is as follows:

msiexec /package <MSI file> <options>

The **/package** option is equivalent to the **/i** option.

To uninstall M-Files, the command line syntax is as follows:

msiexec /uninstall <MSI file> <options>

The **/uninstall** option is equivalent to the **/x** option.

5.2. SILENT INSTALLATION

In most cases, command-line installation will run in silent mode. The remaining of this section assumes that the installation is executed in silent or passive mode. If the installation is executed in full UI mode, some of the command line options described in this document will not be effective because the choices the user makes in the UI may override the command-line options.

Typically, you would use one of the following command-line options to run the installer in silent or passive mode:

/quiet Quiet display option. The installer runs an installation without displaying a user interface. No prompts, messages, or dialog boxes are displayed to the user. The user cannot cancel the installation.

The **/quiet** option is equivalent to the **/qn** option.

/passive Passive display option. The installer displays a progress bar to the user that indicates that an installation is in progress but no prompts or error messages are displayed to the user. The user cannot cancel the installation.

The **/passive** option is equivalent to the **/qb!-** option with REBOOTPROMPT=S set on the command line.

/qb! Basic UI. Similar to **/passive** but may display modal dialog boxes. The **!** character in the option hides the Cancel button.

5.3. LOGGING

You should enable logging especially if the installation fails or does not produce the expected results.

Typically, you would use one of the following command-line options to enable logging:

/log <log file> Writes logging information into a log file at the specified existing path. The path to the log file location must already exist. The installer does not create the directory structure for the log file.

The **/log** option is equivalent to the **/l*** option. To include "verbose output" and "extra debugging information", specify the **/l*vx** option.

5.4. FEATURES TO INSTALL

By default, if you run the installer in silent or passive mode, the setup program installs only the Client feature. The Server and ServerTools features are not installed.

Note: In the M-Files Setup user interface, the features are called "components" and have the following display names. In Windows Installer terminology they are "features", and for clarity this section uses the Windows Installer terminology.

Client = "M-Files Desktop"
Server = "M-Files Server"
ServerTools = "M-Files Server Tools"

There are two ways to control the selection of features: specifying the INSTALLLEVEL property, or by specifying individual features with the ADDLOCAL and REMOVE properties.

5.4.1 INSTALLLEVEL

The INSTALLLEVEL property is effective for new installations only. It controls the selection of features to install by causing all features with an "install level" smaller than or equal to the specified INSTALLLEVEL to be installed. The features in the M-Files installation package have the following install levels:

Support	1
Client	3
ServerTools	20
Server	30

The Support feature must always be installed. It is automatically included in installation whenever any of the other features is selected. The Support feature includes M-Files API and other common modules.

By default, the INSTALLLEVEL property of the M-Files installation package is 3, which causes only the Client feature to be installed (as well as the mandatory Support feature). This is the default behavior if the installer is run in silent or passive mode and no INSTALLLEVEL is specified on the command line.

To install Client and ServerTools, specify **INSTALLLEVEL=20** on the command line.

To install Client, ServerTools, and Server, specify **INSTALLLEVEL=30** or a higher value such as INSTALLLEVEL=100 on the command line.

To install only the Support feature, e.g., for an M-Files API only installation, specify **INSTALLLEVEL=1** on the command line.

Example:

msiexec /package M-Files_x64_eng_11_0_4210_0.msi /quiet INSTALLLEVEL=20

Installs M-Files Desktop and M-Files Server Tools (M-Files Admin) in a completely silent mode.

5.4.2 ADDLOCAL

Use the ADDLOCAL property to specify a list of features, delimited by commas, to be installed. In a new installation, only the specified features will be installed. If the same version is already installed, the features specified in the ADDLOCAL property will be installed in addition to the already installed features.

You do not need to specify the Support feature in the ADDLOCAL property. The Support feature is automatically included whenever any of the other features is installed.

To install Client only, specify **ADDLOCAL=Client** on the command line. This is the default behavior if the installer is run in silent or passive mode and no ADDLOCAL property is specified on the command line.

To install Client and ServerTools, specify **ADDLOCAL=Client,ServerTools** on the command line.

To install Server and ServerTools only, specify **ADDLOCAL=Server,ServerTools** on the command line.

To install Client, ServerTools, and Server, specify **ADDLOCAL=Client,ServerTools,Server** or **ADDLOCAL=ALL** on the command line.

Example:

msiexec /package M-Files_x64_eng_11_0_4210_0.msi /passive ADDLOCAL=Client,ServerTools

Installs M-Files Desktop and M-Files Server Tools (M-Files Admin), showing a progress dialog to the user but with no ability for the user to cancel the installation.

5.4.3 REMOVE

Use the REMOVE property to specify a list of features, delimited by commas, to be uninstalled.

To remove the Client feature from an existing installation, specify **REMOVE=Client** on the command line.

Example:

msiexec /package M-Files_x64_eng_11_0_4210_0.msi /passive REMOVE=Client

Uninstalls the M-Files Desktop component from an existing M-Files installation.

5.4.4 M_REMOVE_ALL_PREVIOUS_VERSIONS

Use M_REMOVE_ALL_PREVIOUS_VERSIONS property to remove previous M-Files versions from a user's workstation. Note that only three older versions will be removed. Removed versions are selected from the latest to the oldest. E.g., if there is M-Files versions 8, 9, 10, and 11, the version 8 is not removed.

Example:

msiexec /package M-Files_x64_eng_12_0_4522_0.msi M_REMOVE_ALL_PREVIOUS_VERSIONS="1"

Uninstalls three previous M-Files versions.

5.5. DESTINATION FOLDER

Use the INSTALLDIR property to specify the destination folder for the installation.

In a new installation, M-Files is installed to a folder named "M-Files" under the system's "Program Files" location by default (e.g., "C:\Program Files\M-Files"). A sub-folder corresponding to the M-Files version is created under this folder.

In an upgrade installation, M-Files is installed to the same folder as the previous version by default. For example, if the previous version is 10.0.3911.85 and is found in "D:\M-Files\10.0.3911.85", the setup program will set the INSTALLDIR property to "D:\M-Files" by default and the new version will be installed in "D:\M-Files\<new version>".

In both a new installation and an upgrade installation, you can override the default destination folder by specifying the INSTALLDIR property on the command line.

Example:

msiexec /package M-Files_x64_eng_11_0_4210_0.msi /passive INSTALLDIR="C:\My Apps\MF"

Installs M-Files to a folder named "C:\My Apps\MF".

5.6. NEW VS. UPGRADE INSTALLATIONS

The default behavior of M-Files Setup depends on the presence of other versions of M-Files on the computers.

If a **newer** version of M-Files is already installed on the computer, then by default the installation will terminate without installing the older version if the installation is run in quiet or passive mode. This helps avoid accidentally installing an older version of M-Files via centralized deployment by for example a Group Policy object (GPO). To install an older version of M-Files on a computer that already has a newer version installed, run M-Files Setup in full UI mode or specify M_CAN_INSTALL_OLDER=1 on the command line.

If the **same** version of M-Files is already installed on the computer, the installation will terminate without modifying the system.

If an **older** version of M-Files is already installed on the computer, then by default the installation will upgrade the previous version to the new version ("upgrade installation"). Only the features that were previously installed will be upgraded. For example, if the previous version has the Client and ServerTools features installed, then at the end of the

upgrade installation with default options ("simple upgrade"), the new version will have the Client and ServerTools features installed and the settings and data from the previous version have been automatically migrated to the new version. The installer will automatically remove the installation of the previous version.

The default functionality can be overwritten with the M_UNINSTALL_PREVIOUS_CLIENT="0" and M_SIMPLE_UPGRADE="0" parameters. When you use these parameters together, the old version is not removed.

Example: msiexec /package M-Files_Online_x64_eng_20_3_8876_7_EV.msi M_UNINSTALL_PREVIOUS_CLIENT="0" M_SIMPLE_UPGRADE="0" /quiet

If needed, you can also prevent the copying of the settings from the older version with the M_MIGRATE_CLIENT="0" parameter. You can modify both parameters by replacing CLIENT with SERVER, SERVERTOOLS, or COMMON, depending on what you are installing.

If **no** version of M-Files is already installed on the computer, a new installation is performed.

5.7. M-FILES DESKTOP DRIVE LETTER

> **Note**: The instructions in this section apply only when the end user is installing M-Files. These do not apply when the administrative user is distributing M-Files to client computers.

In a new installation of M-Files, the drive letter for the virtual M-Files drive is M: by default, or if the M: drive letter is already reserved, the first free drive letter from Z towards A.

To specify a custom drive letter for the virtual M-Files drive in a new installation, specify **M_MFCLIENT_DRIVE_LETTER=X** on the command line, where X is the desired drive letter. This forces the virtual M-Files drive on the selected drive even if there is a network drive, or such, already installed.

Alternatively, with M-Files versions 2015.2 and later, you can specify a range of drive letters for the virtual M-Files drive. Specify **M_MFCLIENT_CUSTOM_DRIVE_LETTERS=XYZ…** on the command line where X, Y and Z are the desired drive letters in order in which they should be tried.

These options are not effective in an upgrade installation. An upgrade installation always preserves the same drive letter that was used with the previous version.

6. CENTRALIZED DEPLOYMENT VIA GROUP POLICY

You can use the Group Policy feature of Windows to automatically distribute M-Files to client computers. Alternatively, you may use any other centralized deployment mechanism that you are familiar with. This section provides instructions on how to create a Group Policy object (GPO) that automatically installs M-Files Desktop to the specified client computers.

The following characteristics of the M-Files installation package make M-Files well suited for Group Policy deployment:

- When run in silent mode with default options, M-Files Setup installs the Client feature only.
- The installation package can be easily customized to include the organization's vault connections, enabling you to encapsulate the vault connections configuration into the MSI package.

- If the target computer already has a newer version of M-Files installed, the setup program will terminate without installing the older version on such a computer. This helps avoid unintended installations of an older version of M-Files to computers that already have a newer version installed.
- If the target computer already has an older version of M-Files installed, the previous version's installation will automatically be upgraded to the new version, preserving all settings and data from the previous version, and the previous version will get uninstalled.

In order to automate the deployment of M-Files via Group Policy, you should be familiar with the basic principles of the Group Policy feature in Windows. See Group Policy in Microsoft Learn for more information on Group Policy in general.

This document describes one possible approach for deploying M-Files Desktop to users' computers via Group Policy.

6.1. CREATING THE GROUP POLICY OBJECT (GPO)

Use the **Group Policy Management** console (GPMC, gpmc.msc) to create a new Group Policy object (GPO). GPMC is typically already installed on domain controllers. You can also install Group Policy Management Tools that are part of Remote Server Administration Tools (RSAT) on a workstation.

Follow these steps to create a new Group Policy object (GPO) in GPMC:

1. Open the **Group Policy Management** console (gpmc.msc).
2. Navigate to the **Group Policy Objects** node under your forest and domain.
3. Right-click the Group Policy Objects node and choose **New** to create a new unlinked GPO.
4. Enter **"Install M-Files Desktop"** as the name for the new GPO and click OK.
5. Select the "Install M-Files Desktop" GPO in the tree view.
6. Right-click the "Install M-Files Desktop" GPO and choose **Edit**. The Group Policy Management Editor console appears.

In the Group Policy Management Editor console, perform these steps to add the M-Files installation packages to this GPO:

1. Go to the node **Computer Configuration / Policies / Software Settings / Software installation**.

   Adding the installation package to the GPO:

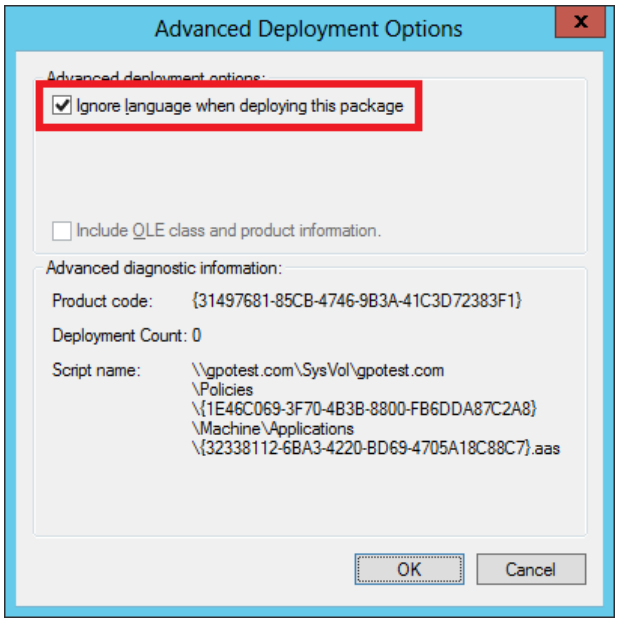
2. Right-click the Software installation node and choose **New / Package**.
3. Point to the installation package of M-Files via a network path (UNC path). If you are using the default M-Files installation packages, the name of the MSI file is similar to "M-Files_x64_eng_11_0_4210_0.msi".
   Note: It is critical this network path is accessible by all computers on which you intend the GPO to apply. In practice, this should be a shared folder and the **Authenticated Users** group should have read access to the shared folder and its content.
4. Select the **Advanced** deployment mode and click OK. Adding the MSI package will take several seconds, after which the Properties window for the package will appear.
5. In the Properties window for the package, do these configuration steps:

a. On the General tab, add **" - via GPO"** to the **Name** field. The name should read, for example, "M-Files 11.0.4210.0 - via GPO". The "via GPO" suffix will appear in the Programs and Features listing on target computers, making it easy to see that the application has been installed via a Group Policy object (GPO).

b. On the Deployment tab, click **Advanced**.

c. Turn **ON** the option **"Ignore language when deploying this package"**. This ensures that the package gets deployed even if the target computer's language does not match the package language.

d. The Advanced Deployment Options dialog box should look as follows:



e. Click OK to close the Properties window of the package.

6. Click OK to close the Properties window of the package.

7. Close the Group Policy Management Editor window.

You have now created the Group Policy object (GPO). However, it has not yet been linked to any domain and thus is not yet effective.

6.2. DISABLING THE GROUP POLICY OBJECT (GPO)

To prevent the GPO from accidentally applying to a larger scope than intended, you should disable the GPO before linking it to any site, domain, or organization unit node. To disable the GPO, right-click the GPO and choose **GPO Status / All Settings Disabled**.

6.3. SPECIFYING THE TARGET SCOPE FOR THE GPO

In order to have the Group Policy object (GPO) apply to one or more computers, you need to link it to a site, domain, or organizational unit node. Additionally, you may want to use Security Filtering or WMI Filtering to further limit the scope to which the GPO will apply.

6.3.1 LINKS

Linking a GPO to a node in the Active Directory hierarchy is one of the three ways to control the **scope** to which the GPO applies. This aspect of the scope definition is mandatory: if you do not link the GPO to any node, it will not apply to any computers. The links are shown in the **Links** section of the display when the GPO is selected in the Group Policy Management console.

If you have e.g. an organizational unit "Headquarters/Desktops" that groups all workstations in your headquarters location, you could select that node and use the **Link Existing GPO** command to link the previously created "Install M-Files Desktop" GPO to the "Headquarters/Desktop" node. This will cause M-Files Desktop to be automatically installed to all computers that belong to the "Headquarters/Desktops" organizational unit.

You should typically **not** link the GPO directly to the domain node, unless you are using the Security Filtering feature to control where the GPO applies. Otherwise, with the default security filtering settings, a GPO that is linked directly to the domain node will apply to all computers in the domain, including the domain controller and other servers. This is often not the intention when deploying M-Files Desktop via Group Policy. Instead, you should link the GPO to a node that contains only workstation computers in your Active Directory hierarchy. You can use the **Active Directory Users and Groups** console to view your Active Directory hierarchy and organize computers into organizational units.

6.3.2 SECURITY FILTERING

The second way of controlling where the GPO should apply is to use the **Security Filtering** settings of the GPO. By default, Security Filtering allows the GPO to apply to all **Authenticated Users** (which includes all computer accounts in the domain). To limit the GPO to apply to only a specific set of computers, you can remove the Authenticated Users entry from the Security Filtering section of the GPO and add individual computer accounts or groups instead.

6.3.3 WMI FILTERING

The third way of controlling how widely the GPO applies is to use WMI Filtering. WMI Filtering enables advanced control such as applying the GPO only to computers with a specific OS version, or specific hardware configuration. WMI Filtering is typically more complex to specify and manage than the other two ways of controlling the GPO's reach, so in most cases you should rely on the **Links** and **Security Filtering** settings instead.

6.4. ENABLING THE GPO

Once you have linked the GPO to a site, domain, or organizational unit node and, if necessary, adjusted Security Filtering to control the scope to which the GPO applies, you can enable the GPO by selecting the GPO under the Group Policy Objects node and selecting **GPO Status / Enabled** from the right-click menu.

6.5. VERIFYING THE RESULTS

The GPO will cause M-Files Desktop to be automatically installed on the target computers when the computers are restarted the next time (assuming that the GPO has been applied by then). To ensure that the GPO gets applied, you can run the **gpupdate /force** command on one of the target computers and then restart the computer. This ensures that the GPO gets applied during the restart.

The GPO causes M-Files Desktop to be installed during the next computer restart before any user can log on. Depending on the operating system version, the text "Installing managed software" may appear on screen while M-Files is being installed.

6.6. TROUBLESHOOTING GPO INSTALLATION ISSUES

In case installation via Group Policy does not produce the intended results, see the Windows Event Log on the target computer for additional information.

6.6.1 SYSTEM LOG

In the **System** log, you should find an event that indicates that a Group Policy object (GPO) has been applied:

Source: Application Management Group Policy
Event ID: 301
Description: The assignment of application M-Files 11.0.4210 - via GPO from policy Install M-Files Desktop succeeded.

The following events in the **System** log indicate that the installation has succeeded:

Source: Application Management Group Policy
Event ID: 302
Description: The install of application M-Files 11.0.4210.0 - via GPO from policy Install M-Files Desktop succeeded.

Source: Application Management Group Policy
Event ID: 308
Description: Changes to software installation settings were applied successfully.

Source: GroupPolicy
Event ID: 1502
Description: The Group Policy settings for the computer were processed successfully. New settings from 1 Group Policy objects were detected and applied.

6.6.2 APPLICATION LOG

If the GPO is getting applied, you should see events similar to the following in the **Application** log:

Source: MsiInstaller
Event ID: 1040
Description: Beginning a Windows Installer transaction: {d57ec893-e41c-47ee-bc6c-43921c4228e7}. Client Process Id: 856.

Source: MsiInstaller
Event ID: 1042
Description: Ending a Windows Installer transaction: {d57ec893-e41c-47ee-bc6c-43921c4228e7}. Client Process Id: 856.

6.6.3 ENABLING WINDOWS INSTALLER LOGGING FOR GPO INSTALLATIONS

For detailed information on what is happening during the installation, you can enable logging for Windows Installer by specifying the following registry settings on the target computer:

Key name: HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Installer
Value name: Logging
Value type: REG_SZ
Value data: voicewarmupx

Key name: HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Installer
Value name: Debug
Value type: REG_DWORD
Value data: 7

The log file will be created in the currently logged on user's Temp folder and will have a name in the following format: "MSI*.LOG". In case of a GPO deployment, there will be no logged on user at the time the installation occurs. In this case the log file will be created in the "Windows\Temp" folder. Note: This option should not be left active since every install/uninstall operation of an MSI package will create a new log file, thus unnecessarily occupying disk space. Therefore, this option should only be used for debugging purposes.

For more information on Windows Installer logging options in the registry, see the following article: http://support.microsoft.com/kb/2545723

6.6.4 ENSURING SUFFICIENT PERMISSIONS TO THE INSTALLATION PACKAGE FOLDERS

A typical reason for problems in Group Policy deployments is that the target computer cannot access the MSI package. The target computer accesses the UNC path specified in the GPO using the computer account identity DOMAIN\COMPUTER$. If this account cannot read the MSI file, the installation will fail. You can typically identify this situation from the Windows Installer log if you have enabled logging as described in section 6.6.3.

To correct the situation, ensure that the GPO is referring to the MSI package via a UNC path that is valid and accessible from the target computers. Further, ensure that the **Authenticated Users** group has read access to the UNC path and its content. You may need to grant the Authenticated Users group access also to the parent folders of the shared folder that contains the MSI file.

7. UPGRADING M-FILES VIA GROUP POLICY

If you have already deployed a version of M-Files via Group Policy as described in section 6 and a new version of M-Files becomes available, you can follow the instructions in this section to modify the previously created GPO to deploy the new version as an upgrade to existing targets and as a new installation to new computers.

There are several different ways that you can use to deploy an M-Files upgrade via Group Policy. This section describes the approach of adding the new version's installation packages to the same GPO that was previously created. This approach has the advantage of keeping the number of GPOs small. Another possibility is to create a separate GPO for the new version's deployment.

In general, deploying an M-Files upgrade via Group Policy does not significantly differ from deploying the first installation. When applied to a target computer, the M-Files installation package will automatically detect if the computer has an older version of M-Files installed. If yes, the new version will automatically run in "simple upgrade" mode and will migrate all settings and data from the previous version to the new version. At the end of a successful installation, the previous version will get uninstalled.

This means that the GPO does not need to include any steps for uninstalling the previous version. In fact, it is important that the GPO does **not** try to uninstall the previous version. Instead, the GPO must leave the migration and uninstallation logic to the responsibility of the M-Files setup program.

7.1. ADDING A NEW VERSION OF M-FILES TO THE GPO

Use the **Group Policy Management** console (GPMC, gpmc.msc) to edit the previously created Group Policy object (GPO):

1. Open the **Group Policy Management** console (gpmc.msc).
2. Navigate to the **Group Policy Objects** node under your forest and domain.
3. Select the **"Install M-Files Desktop"** GPO in the tree view.
4. Right-click the "Install M-Files Desktop" GPO and choose **Edit**. The Group Policy Management Editor console appears.

In the Group Policy Management Editor console, perform these steps:

1. Navigate to the node **Computer Configuration / Policies / Software Settings / Software installation**.
2. Right-click the Software installation node and choose **New / Package**.
3. Add the installation package of the new M-Files version with the same steps as described in section 6.1. **Important:** Follows the steps in section 6.1 carefully.
4. In addition to the steps listed in section 6.1, activate the **Upgrades** tab in the Properties window of the new package. Use the **Add** button to add the previous M-Files version's package from the "Package to upgrade" list. Select the **"Package can upgrade over the existing package"** option and click OK.

   IMPORTANT: Make sure to select the **"Package can upgrade over the existing package"** option. If you leave the default option "Uninstall the existing package, then install the upgrade package" selected, the GPO will first uninstall the previous M-Files version from the target computer before installing the new version. This will cause all user settings and in-progress data such as modified checked-out documents on the client computer to be lost. The M-Files setup program has been designed to automatically migrate all settings and data from the previous

version to the new version and then uninstall the previous version. Thus, it is important to leave the uninstalling of the previous version to the responsibility of the M-Files setup program.

5. To prevent the upgrade from being applied to the full scope of the GPO at this time, consider limiting the visibility of the new version's package by changing the settings on the **Security** tab before you click OK to finalize the addition of the package. This approach is described in more detail in section 7.2.
6. Leave the previous M-Files version's installation package in the list. The GPO will only install the latest version in the chain.

7.2. TESTING THE UPGRADE WITH LIMITED SCOPE

Using the original GPO for upgrades simplifies GPO management in many respects but has the downside that the upgrade will be immediately applied to the full scope of the GPO. You can avoid this by using package-specific security settings. When adding the new M-Files version's installation package to the GPO, you can adjust the settings on the **Security** tab to limit the visibility of the new version's package to specific test computers only.

By default, the security settings for the package in the GPO grant read access to the full scope of the GPO. To limit the visibility of this individual package, click the **Advanced** button.

In the Advanced Security Settings dialog box, click **Disable inheritance** and select the **Remove all inherited permissions from this object** option. Then add the desired test computer accounts to the list individually by clicking the **Add** button. The default permissions for the added entry will grant read access, which makes the package apply to that computer.

With these settings, the new version's package will not apply to the full scope of the GPO yet. You can verify that the GPO works as intended by running **gpupdate /force** on one of the test computers to which you applied the new version's package and by restarting the computer.

Once you have completed testing with the limited scope of one or more test computers, you can change the permissions of the new version's package in the GPO back to the default inherited permissions. In Advanced Security Settings, click **Enable inheritance** and click **Apply**. Then, for clarity, remove all the non-inherited permission entries (the ones with "Inherited from" showing "None" in the Permission entries list).

8. DISTRIBUTING VAULT-SPECIFIC CLIENT SETTINGS

There are various client-side settings that can be used for controlling the behavior of M-Files Desktop. Many of these settings are ***vault-specific***. This means that they can be used in one vault but not in others. This is especially true if the settings refer to the metadata structure of the vault, such as settings that control which property definitions should be used in automatic filling in the metadata card.

The vault-specific client settings are primarily controlled in M-Files Admin through **Advanced Vault Settings** > **Client**. When a user logs in to the vault, M-Files Desktop reads the settings from the vault on the server and applies the settings to the client computer's registry. This ensures that any user that accesses the vault will use the same settings.

These vault-specific settings are stored in the HKEY_CURRENT_USER (HKCU) subtree under the following keys:
`HKEY_CURRENT_USER\Software\Motive\M-Files\<version>\Client\Common\<vault>`
`HKEY_CURRENT_USER\Software\Motive\M-Files\<version>\Client\MFShell\<vault>`

This mechanism is applicable only for vault-specific client settings. For ways to distribute other types of client-side settings, see sections 3.2.5 and 9.

9. DISTRIBUTING CLIENT-SIDE REGISTRY SETTINGS VIA GROUP POLICY

There are various client-side registry settings that can be used for controlling the behavior of M-Files Desktop. Some of these preferences are available as options in M-Files Desktop Settings on the Settings tab, and the value of the setting is stored in the computer's registry. Some other settings are only available via a registry setting. In both cases you may want to distribute the desired registry settings to all client computers in a centralized way. Group Policy object (GPO) in Active Directory is a good way of achieving this.

For example, in order to change the auto-closing timeout of the logout confirmation message box from the default 15 seconds to 30 seconds for all users on all computers in your organization, you would distribute the following registry setting to all client computers via a GPO:

Key name: HKEY_CURRENT_USER\Software\Motive\M-Files\%MFilesVersion%\Client\Common
Value name: CheckedOutWarningAutoCloseInSeconds
Value type: REG_DWORD
Value data: 30

For distributing vault-specific settings via Advanced Vault Settings, see section 8.

9.1. THE MFILESVERSION SYSTEM ENVIRONMENT VARIABLE

M-Files registry settings are version-specific: the version of M-Files appears as a key under the Software\Motive\M-Files key. For example, all registry settings for M-Files 11.0.4210.0 would appear under the following key name: "Software\Motive\M-Files\11.0.4210.0".

In order to make it easier for you to distribute registry settings via a GPO without necessarily knowing the version of M-Files that is installed on a particular target computer, M-Files Setup sets a system environment variable named **MFilesVersion** when M-Files is installed. The value of the MFilesVersion system environment variable contains the M-Files version string such as "11.0.4210.0". The MFilesVersion variable is supported in M-Files 11.0.4207.0 and later.

If the computer has multiple versions of M-Files installed, the MFilesVersion variable refers to the newest version on the computer (regardless of the order in which the versions have been installed). For example, if you have installed versions 11.0.4250.0 and 11.0.4210.0 on the same computer, the value of the MFilesVersion system environment variable is "11.0.4250.0". Of course, in most production-use environments you would expect users to have only a single version of M-Files installed.

Group Policy objects (GPOs) support the use of environment variables in key names when you specify registry settings via GPO. Thus, you can specify the key name as **"Software\Motive\M-Files\%MFilesVersion%"** to target whatever versions the users have installed on their computers. When the M-Files software is upgraded to a newer version, you will not need to change the key names of the registry items in the GPO.

9.2. DISTRIBUTING HKEY_CURRENT_USER REGISTRY SETTINGS VIA GPO

Many of the client-side registry settings of M-Files are located in the HKEY_CURRENT_USER (HKCU) subtree. Because the HKCU settings are user-specific, your Group Policy object (GPO) should be targeted at users instead of computers. Thus, it is best to create a separate GPO for the HKCU settings instead of including them in the same GPO that contains the M-Files installation packages.

Create a new GPO similar to how a GPO was created in section 6.1. You could target this GPO to all users in your domain whether or not they will be using M-Files. The M-Files-specific registry settings will not have any effect on the user's computer if the M-Files software is not installed: the registry values will be present in the user's HKCU subtree but no application other than M-Files will use those settings.

Edit the GPO and add a new registry item as follows:

1. In Group Policy Management Editor, navigate to **User Configuration / Preferences / Windows Settings / Registry**.
2. Right-click the Registry node and choose **New / Registry Item**. The New Registry Properties window appears.
3. Set **Action** to **Update**.
4. Set **Hive** to **HKEY_CURRENT_USER**.
5. Set **Key path** to the name of the registry key, such as the following:
   Software\Motive\M-Files\%MFilesVersion%\Client\Common
6. Set **Value name** to the name of the registry value to set, e.g., "CheckedOutWarningAutoCloseInSeconds".
7. Set **Value type** to the desired type, e.g., REG_DWORD.
8. Set **Value data** to the desired value, e.g. "30". When adding numeric values, you should typically select **Decimal**.
9. Save the registry item by clicking OK.

Verify on a target user's computer that the registry setting is written to the correct location. Run **gpupdate /force** to force an immediate update of the GPOs that apply to the user, and then run **regedit** to verify that the registry settings appears in the registry as expected. If not, verify that the computer has M-Files installed and that the MFilesVersion system environment variable is set to the expected value.

9.3. COMPUTERS THAT DO NOT HAVE M-FILES INSTALLED

If the GPO-distributed registry settings are applied to computers that do not have M-Files installed, the MFilesVersion system environment variable does not exist. In such a case the registry setting will be written to a key name where %MFilesVersion% appears literally in the key path. This does not cause any unwanted side effects on computers that do not have M-Files installed, nor does it cause any problems on computers where the M-Files software is installed. You can simply ignore any registry settings that appear in a key named "Software\Motive\M-Files\%MFilesVersion%", or you can delete them.

If you want to automatically delete registry settings where the string %MFilesVersion% appears literally (i.e., on computers that do not have M-Files installed), you can add the following registry item in the GPO:

Action: Delete
Hive: HKEY_CURRENT_USER
Key path: Software\Motive\M-Files\%<MFilesVersion>% **(IMPORTANT: Note the < and > characters.)**

The syntax "%<MFilesVersion>%" causes the GPO to treat the value as a literal and not expand it to the value of the environment variable even if the environment variable is defined on the target computer.

This registry item should be the last item in the list of registry items in the GPO.

9.4. DISTRIBUTING HKEY_LOCAL_MACHINE REGISTRY SETTINGS VIA GPO

Even though it is possible to package HKEY_LOCAL_MACHINE registry settings into a customized installation package, you may want to distribute also the HKEY_LOCAL_MACHINE (HKLM) registry settings to the users' computers via a Group Policy object (GPO) similar to how you would distribute HKCU settings (see section 9.2). Deploying HKLM settings via a GPO makes sense especially if you need to apply new or modified settings after some users have already installed the M-Files software.

Some of the HKLM registry settings should be available in the registry at the time the M-Files services start for them to be effective. To ensure that this happens even for the first time the M-Files software is installed via GPO and the services get started as part of the installation, you should place the HKLM settings in the same GPO as the software installation packages. However, the MFilesVersion environment variable will not yet have its correct value at the time the registry items in the GPO are applied. Thus, instead of referring to %MFilesVersion% in your HKLM registry items in this GPO, you should add your own system environment variable by specifying one in the Environment section in the GPO and then referring to that variable in the HKLM settings.

For example, add the following system environment variable entry in the GPO's **Computer Configuration / Preferences / Windows Settings / Environment** section:

Action: Update
Type: System Variable
Name: MFilesVersionGPO
Value: 11.0.4250.0

The version string must match the version that the GPO will install.

Then add registry items in the Registry section with values similar to the following:

Action: Update
Hive: HKEY_LOCAL_MACHINE
Key path: Software\Motive\M-Files\%MFilesVersionGPO%\Client\MFClient
Value name: Drive
Value type: REG_SZ
Value data: X

If in addition to this you want to distribute HKLM settings to computers for whatever M-Files version they currently have installed, you can create a separate GPO for those settings and refer to %MFilesVersion% in the key paths of those registry items. Depending on if you place the settings in the Computer Configuration or User Configuration section in the GPO, the settings will take effect at computer restart time or when a user logs on to Windows. If M-Files reads the setting at the time the M-Files services start, the settings will not become effective until after another restart of the computer.

9.5. DISTRIBUTING REGISTRY SETTINGS VIA A BATCH FILE

Another alternative for deploying registry settings on client computers is to use a .cmd or .bat file. The .cmd or .bat file can call the REG command to add registry values. The execution of a .cmd or .bat file resolves any environment variables, meaning that you can use the %MFilesVersion% placeholder in the .cmd or .bat file to refer to the install M-Files version.

The following line in a .bat file writes a value to the HKEY_LOCAL_MACHINE subtree (line breaks have been added for clarity here and should not be part of the actual line in a .cmd or .bat file):

```
REG ADD "HKLM\Software\Motive\M-Files\%MFilesVersion%\Client\MFClient"
        /v AutomaticLogoutTimeoutInMinutes
        /t REG_DWORD
        /d 15
        /f
```

You may distribute registry settings with .reg files as well, but since the .reg files do not support the use of environment variables in key names, you would need to modify the .reg file for each M-Files version.