mofcomp

Article • 01/07/2021 • 6 contributors

In this article

The *Managed Object Format (MOF)* compiler parses a file containing MOF statements and adds the classes and class instances defined in the file to the WMI repository. MOF files are usually automatically compiled during the installation of the systems with which they are provided, but you can also compile MOF files by using this tool.

For more information about locating and using mofcomp.exe, see Using WMI Management Tools. For information about removing classes and instances from the WMI repository, see the **pragma deleteclass** preprocessor command.

The following code example shows how to run the MOF compiler on a file.

syntax Copy

```
mofcomp
  [-autorecover]
  [-check]
  [-N:<namespacepath>]
  [-class:createonly | -class:forceupdate |
   -class:safeupdate | -class:updateonly ]
  [-instance:updateonly | -instance:createonly]
  [-B:<filename>]
  [-WMI]
  [-P:<Password>]
  [-U:<UserName>]
  [-A:<Authority>]
  [-MOF:<path>]
  [-MFL:<path>]
  [-AMENDMENT:<Locale>]
  [-ER:<ResourceName>]
  [-L:<ResourceLocale>]
  <MOFfile>
```

Switches

-autorecover

Adds the named MOF file to the list of files compiled during repository recovery. The list of autorecover MOF files is stored in the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBEM\CIMOM

The MOF files listed in this registry entry must reside on the local computer because MOF files that use the **autorecover** command cannot recover MOF files located on a remote computer.

> ⓘ Note
>
> To ensure that all your WMI class definitions for managed objects are restored to the ***WMI repository*** if WMI has a failure and restarts, use the **#pragma autorecover** preprocessor instruction in your ***Managed Object Format*** (MOF) file.

-check

Requests that the compiler perform a syntax check only and print appropriate error messages. No other switch can be used with this switch. When this switch is used, no connection to Windows Management Instrumentation (WMI) is established and no modifications to the WMI repository are made.

-N:<*namespacepath*>
Requests that the compiler load the MOF file into the namespace specified as *namespacepath*. The compiled MOF is loaded into the default Mofcomp namespace, root\\default, unless this switch is used. You can also insert the preprocessor command **\#pragma namespace ("***namespace path***")** in the MOF file to achieve the same effect. If both the **-N:** switch and the \#pragma namespace command are used, \#**pragma namespace** **autorecover** takes priority. In this case, the only way to compile the MOF into another namespace is to edit the MOF file and change the \#**pragma namespace** command. A remote computer can be specified using \\\\machinename\\root\\default.

-class:createonly

Requests that the compiler not make any changes to existing classes. When this switch is used, the compile operation terminates if a class specified in the MOF file already exists.

-class:forceupdate

Forces updates of classes when conflicting child classes exist. For example, suppose a class qualifier is defined in a child class and the base class tries to add the same qualifier. In **-class:forceupdate** mode, the MOF compiler resolves this conflict by deleting the conflicting qualifier in the child class. If the child class has instances, the forced update fails.

-class:safeupdate

Allows updates of classes even if there are child classes, as long as the change does not cause conflicts with child classes. For example, this flag allows adding a new property to the base class that was not previously mentioned in child classes. If the child classes have instances, the update fails.

-class:updateonly

Requests that the compiler not create any new classes. When this switch is used, the compile operation terminates if a class specified in the MOF file does not exist.

-instance:updateonly

Requests that the compiler not create any new instances. When this switch is used, the compile operation terminates if an instance specified in the MOF file does not exist.

-instance:createonly

Requests that the compiler not make any changes to existing instances. When this switch is used, the compile operation terminates if an instance specified in the MOF file already exists.

-B:<*filename*>

Requests that the compiler create a binary version of the MOF file with the name *filename* without making any modifications to the WMI repository.

If you use the **-B:<*filename*>** option to create a binary MOF file, only default qualifier flavors are stored in the WMI repository.

Binary MOF format is the intermediate format for combining a WDM-driver with the MOF as a resource. The binary MOF represents classes and instances just as a text MOF file does and is compressed before it is stored on disk.

-WMI

Requests that the compiler perform a WMI syntax check. The **-B:** switch must be used with this switch. The **-WMI** switch is only used for building binary MOF files for use by WDM device drivers. This switch invokes a separate binary MOF file checker, which runs after the binary MOF file is created.

-P:<*Password*>

Specifies *Password* as the password for the computer user to enter when logging on.

-U:<*UserName*>

Specifies *UserName* as the name of the user logging on.

-A:<*Authority*>

Specifies *Authority* as the authority (domain name) to use when logging on to WMI.

-MOF:<*path*>

Name of the language neutral output. Used with the **-AMENDMENT** switch to specify the name of the language-neutral MOF file that will be generated.

-MFL:<*path*>

Name of the language specific output. Used with the **-AMENDMENT** switch to specify the name of the language-specific MOF file that will be generated.

-AMENDMENT:<*Locale*>

Splits the MOF file into language-neutral and -specific versions. The MOF compiler creates a language-neutral form of the MOF file that has all amended qualifiers removed. A localized version of the MOF file is also created with an MFL file name extension. The *Locale* parameter specifies the name of the child namespace that contains the localized class definitions. The format of the *Locale* parameter is MS_xxx where xxx is the hexadecimal value of the Windows LCID. For example, the locale for American English is MS_409.

-ER <*ResourceName*>

Extracts binary MOF from a named resource. This switch gets the binary MOF from the class in the WMI repository while the -B switch creates the binary MOF format from a MOF file.

-L:<*ResourceLocale*>

Optional. Extracts the localized MOF descriptions from the binary MOF when used with -ER switch.

<MOFfile>

Name of the file to parse.

Return Values

As its first operation, the MOF compiler performs a syntax check on the MOF file. If the compiler finds any errors, it prints an error message and the process terminates.

The MOF compiler can return the following values:

0

MOF compile operation was successful.

1

The MOF compiler could not connect with the WMI server. This is either because of a semantic error such as an incompatibility with the existing WMI repository or an actual error such as the failure of the WMI server to start.

2

One or more command-line switches were not valid.

3

A MOF syntax error occurred.

If the MOF file is parsed correctly, but an attempt is made to perform an operation that is forbidden by a command-line switch, the compiler returns an error code generated by WMI instead of any of the return codes listed in the list preceding. For example, a WMI error code is returned when the **-instance:updateonly** switch is specified and the MOF file attempts to create an instance.

If the **#pragma autorecover** preprocessor statement is not in the file, then the following warning is returned:

syntax Copy

```
WARNING: FileYourMof.Mof does not contain #PRAGMA AUTORECOVER.
If the WMI repository is rebuilt in the future, the contents of this
MOF file   will not be included in the new WMI repository.
To include this MOF file when the WMI Repository is automatically
reconstructed, place the #PRAGMA AUTORECOVER statement on the first
line of the MOF file.
```

Remarks

The MOF Compiler is available in the %Windir%\System32\wbem directory. You must specify the MOF file as the parameter of the MOF Compiler. You can also specify an Autorecover switch if you want the MOF file to be automatically recompiled if the CIM Repository ever has to be automatically recovered. For more information, type **Mofcomp /?** at the command prompt.

A MOF file that uses the Unicode character set contains a signature as the first two bytes of the file. This signature is either U+FFFE or U+FEFF, depending on the byte ordering of the file.

When no errors occur in the parsing process, the MOF compiler connects to the WMI server running on the local computer unless the **-check** switch is specified. Classes and instances defined in the MOF file are added to the WMI repository.

When an error occurs in updating the WMI repository, the compiler makes no attempt to return the repository to its state before the compiler began processing.

Windows 8: When installing a provider, mofcomp treats the [Key] and [Static] qualifiers as true if they are present, regardless of their actual values. Other qualifiers are treated as false if they are present but not explicitly set to true.

Requirements

Expand table

Requirement	Value
Minimum supported client	Windows Vista
Minimum supported server	Windows Server 2008

See also

pragma namespace

Compiling MOF Files

Compiling Localized MOF Files

Registering a Provider

IMOFCompiler::CompileFile

Feedback

Additional resources

Training

Module

Query and manipulate repository objects by using CIM and WMI methods - Training

This module explains how to use CIM and WMI to make changes by using methods. The methods available vary depending on the type of object. Discovering and understanding these methods is an important step in querying and manipulating the repository information.

Events

Nov 20, 12 AM - Nov 22, 12 AM

Gain the competitive edge you need with powerful AI and Cloud solutions by attending Microsoft Ignite online.

Register now