# WuInstall

# Table of contents

# New in WuInstall 2.0

## The following options are available in WuInstall 2.0 only

/reboot_if_needed_force
/clearlocalcache
/releasedate
/proxyauto
/proxyaddress
/proxyuser
/proxypass
/targetgroup
/offlinescan
/options
/tempdir

# WuInstall Options

## Contents

# Basic Options

**Contents**

## /help

This option displays the usage.

## /search

Searches for available updates using the windows update API. If no criteria (see /criteria) is specified, it searches for not already installed updates (IsInstalled=0 and Type='Software'). Drivers are not included in the default /search option!

If the use of a WSUS is configured (this is the case when HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU\UseWUServer is set to 1), it uses the configured WSUS Server (which is set with HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\WUServer\) to get the list of updates matching the criteria. Otherwise (when UseWUServer is set to 0), it goes directly to the Microsoft Update Site.

If there is a WSUS Server configured, WuInstall will report for example:

```
You are using WSUS Server http://wsus.mycompany.local:8530
```

Otherwise, when going to the Microsoft Update Site, it will report:

```
You have no WSUS Server configured
```

It is possible to temporarily set, reset or change the use of a WSUS for the execution time of WuInstall, see /bypass_wsus and /use_wsus.

The search for updates will start with the message

```
Searching for updates ... Criteria: IsInstalled=0 and Type='Software'
```

and will ask either the WSUS or the Microsoft Update Site for a list of available updates matching the criteria. If the criteria is invalid, WuInstall will print "Invalid criteria" and exits here, returning code 4.

Sometimes, the search might fail for some reason, in case of an error WuInstall tries the search automatically a second time. If the search fails once more, WuInstall exits with return code 1 and also displays the error code for the reason why the search failed:

```
Search failed ... retrying ...Search failed 2 times.
Result CODE: 0x0 0x8024402C - The proxy server or target server name cannot
be resolved.
```

If the search was successful, WuInstall types:

```
Result Code: Succeeded
```

If no updates were found, it will report:

```
No Updates are available, nothing to do!
```

and exit with return code 2.

If updates were found, it will display the total number of updates found:

```
10 Updates found in total, filtering ...
```

Now, the updates are filtered, if there are filters specified. For details on filtering see the following options: /severity, /classification, /product, /match, /nomatch, /matchfile, /nomatchfile, /isdownloaded

If no update matches the filters, WuInstall will report:

```
No Updates available for specified match string / severity / product!
```

and exit with return code 3.

When there are updates which match the filters they are displayed in a result list like this:

```
1.  Definition Update for Microsoft Forefront Client Security - KB977939
(Definition 1.87.1797.0)
 Severity: Unknown
 UpdateClassification - Definition Updates
 Product - Forefront Client Security
```

The first line contains the position and the complete name of the update, afterwards the severity, classification and product are displayed.

When using the /search option, after listing the updates, WuInstall exits with code 0, without any further action.

only /search was specified, stopping here!

So this option is useful for just checking which updates are available (and, for example, with the criteria "IsInstalled=1" also for checking which updates are already installed) or for testing search criterias and filter options before downloading and installing.

## /download

The /download option first of all executes a search (exactly the same one the /search option does, considering all parameters for criterias and filters).

If there are entries in the result list (updates, matching the criterias and filters, were found), it immediately starts downloading them (either from WSUS or the Microsoft Update Site), displaying:

```
Downloading updates ...
```

The updates are downloaded in exactly the same order in which they are displayed after the search.

When starting the download, WuInstall displays:

```
Downloading Definition Update for Microsoft Forefront Client Security -
KB977939 (Definition 1.87.608.0):
```

It is possible to track the download progress (see the /show_progress option).

Updates, which already have been downloaded (they can be excluded in the result list of the search with the /isdownloaded option), are usually not downloaded again, so the download is "finished" immediately.

WuInstall usually (depending on OS version and system configuration) caches the downloads in the directory Windows\SoftwareDistribution\Download. Using the /download_to option instead of /download, it is also possible to put exe / msi of the downloaded updates to a certain directory for manual installation.

When the download of an update fails for some reason, WuInstall displays

```
Download failed!
```

and the error code and reason why the download failed. WuInstall then immediately exits with return code 1 without downloading the subsequent pending downloads in the result list.

When all updates in the result list have been successfully downloaded, WuInstall displays

```
Downloading finished, listing downloaded updates and counting...
Result Code: Succeeded
```

```
only /download was specified, stopping here
```

and exits with return code 1.

## /download_to_directory

It does almost exactly the same as /download, but furthermore this option copies the updates to the specified directory, usually in msi or exe format.

These updates can be installed manually afterwards.

After downloading an update to the update cache (usually Windows\SoftwareDistribution\Download), WuInstall displays for example

```
Copying download to c:\my_updates...
```

and displays an "OK" if successful, otherwise an error code.

Most updates are update bundles, which means there are multiple sub-updates, each will create a separate installation file in the directory.

WuInstall will display for each bundle:

```
Copying bundle .NET 4 Client x86 (Win7)  ....  OK
```

With update bundles it is quite likely to have more installation exe / msi files than updates.

If copying an update file to a directory fails, WuInstall just displays the error code like this:

```
Copying bundle Microsoft .NET Framework 4 Client Profile Language Packs x86
.... Error: 0x80246007
```

but it does not force WuInstall to abort the entire process, it continues with the next update in the result list (or the next sub-update in the bundle) in any case.

The option /download_to can also be used to copy updates which are already in the download cache to a directory (with /isdownloaded 1).

## /install

This option first executes exactly the same as /download (=searching and downloading updates) and then starts installing the downloaded updates.

Before installation, the downloaded updates are listed and checked if they have been downloaded properly and if they are ready to be installed. Furthermore, it is listed if the Eula is known ("Eula Accepted: true") - otherwise, see /autoaccepteula option.

This looks for example like this:

```
Install was specified. Beginning Install
  Portuguese (Brazil) Language Pack - Windows 7 (KB972813), Eula Accepted:
true
  Danish Language Pack - Windows 7 (KB972813), Eula Accepted: true
1 Updates are ready to be installed!
Beginning install ...
```

Now, the installation starts, all updates are installed in one single step. It is possible to track the installation progress (see the /show_progress option).

If the installation fails for some reason, WuInstall prompts

```
Install failed!
```

and immediately exits with code 1 and shows the error code.

In case the installation ran through, WuInstall enumerates the result for every single update and finally the overall result code:

```
Enumerating result for every single update
 1. Portuguese (Brazil) Language Pack - Windows 7 (KB972813): Result Code:
Succeeded
 2. Danish Language Pack - Windows 7 (KB972813): Result Code: Succeeded
Overall Result Code: Succeeded
```

Afterwards, it is checked if one or more updates need a reboot.

```
Install or Download finished, reboot required: NO
```

If yes, WuInstall exits with code 10, otherwise with code 0.

## /uninstall

This option uninstalls updates from a machine.

The /uninstall option automatically searches for already installed updates offline (see /offline option) with the criteria "IsInstalled=1" and uninstalls them. All filter options (e.g. /match, /product, /severity, ...) are available to specifically select updates to be uninstalled.

Typing just WuInstall /uninstall without any filter options is not advised!

For example, to uninstall KB2251419 use:

**WUInstall.exe /uninstall /match KB2251419**

```
You are using WSUS Server http://mywsus.local:8530

Searching offline for already installed updates...

Searching for updates ... Criteria: (IsInstalled=1)
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing those matching search expression:  'KB2251419'


1. Security Update for Microsoft Office Word 2007 (KB2251419)
 Severity: Critical
 Product - Office 2007
 UpdateClassification - Security Updates

Uninstalling Updates ... Successful
Enumerating result for every single update
```

```
1. Security Update for Microsoft Office Word 2007 (KB2251419): Result Code:
Succeeded

Overall Result Code: Succeeded
```

The /uninstall option only works with updates which have been installed using a WSUS.

Otherwise it usually gives the error code 0x80240028 - WU_E_UNINSTALL_NOT_ALLOWED (The update could not be uninstalled because the request did not originate from a WSUS Server).

There are also some updates which fail to be uninstalled with WuInstall because of internal dependencies.

## Reboot / Shutdown Options

### Contents

### /reboot [nseconds]

This option initiates a system reboot with a delay of the specified number of seconds (nseconds) after the execution of WuInstall.

If no time is specified, 10 seconds are used by default.

At the start of WuInstall execution, the message

```
/reboot specified, system reboots after WuInstall with 10 seconds delay
```

is displayed. WuInstall is executed according to the specified parameters (also, just WuInstall /reboot alone is possible), and after the execution of WuInstall, the message

```
Rebooting after 10 seconds ...
```

is displayed. WuInstall exits with return code 5 and the system will reboot after the specified delay in ANY CASE, regardless of the outcome of WuInstall.

If initiating the reboot fails for some reason, the error code is displayed

```
Reboot failed, Error Code: 1271
```

and WuInstall exits with code 6.

### /reboot_if_needed [nseconds]

Does exactly the same as /reboot, the only difference is that a reboot is triggered only when a reboot is required (when it exits with a return code 1 or 11).

If a reboot is needed, the /reboot_if_needed options generates the output

```
reboot_if_needed specified, rebooting
```

If no reboot is needed, WuInstall puts:

```
no reboot needed, so system will not reboot ...
```

### /rebootcycle [ntimes]

With the /rebootcycle option it is possible to install updates with several reboots in between.
After a successful update and reboot, WuInstall starts and executes the previous command again. If the option is set WuInstall prints:

```
/rebootcycle specified, going to run this wuinstall command at most x times
again at startup if reboot is triggered
```

You can use this feature together with the options /reboot and /reboot_if_needed. If none is specified, /reboot_if_needed is taken as default setting. You are also able to set the maximum amount of reboot cycles (ntimes), for example

**wuInstall /install /reboot_if_needed /rebootcycle 4**

The default value of maximum reboot cycles, if nothing for ntimes is specified, is 3. WuInstall uses the Windows Task Scheduler for executing commands before the user login.

### /reboot_force [nseconds]

Forces a reboot after executing WuInstall after nseconds (or default 10) seconds, it does not matter if a reboot is needed or not. Open programms are forced to be closed.

### /reboot_if_needed_force [nseconds]

**Available in WuInstall 2.0**

This option is a combination of the options /reboot_if_needed and /reboot_force.

### /shutdown [nseconds]

/shutdown is similar to /reboot, but instead of rebooting the system, it shuts down. The default number of the the delay seconds is 10, and the shutdown is done in ANY CASE, regardless the result of WuInstall.

At the beginning of WuInstall it is displayed:

```
/shutdown specified, system system shuts after WuInstall with 10 seconds delay
```

And when WuInstall has finished

```
Shutdown after 10 seconds ...
```

If the shutdown was initiated successfully, WuInstall exits with code 5, otherwise it displays an error code and exits with code 6.

## Caching Options

**Contents**

## /install_cache cache_dir

This feature uses a shared cache directory to prevent machines in a local network from downloading updates more than once.

The /install_cache option first of all executes the online search (exactly the same one as the /search option does, considering all parameters for criterias and filters).

The parameter cache_dir should be a folder on a local network share where multiple machines have access to.

Before downloading the updates from the result list, WuInstall checks in the cache_dir if the update has already been downloaded there by another machine. If the update is not the cache folder, WuInstall downloads the update (from the configured download location - WSUS and/or windowsupdate.com) and puts it into the cache directory, in a subfolder which has the same name as the Update ID.

If the update is present in the folder (if there is a subdirectory with the Update ID), WuInstall tries to copy the update from this shared cache folder to the local update cache. If this fails for some reason, WuInstall deletes the subdirectory in the shared cache folder (because there is apparently something wrong with the update) and downloads the update directly from the Internet.

For example, the first machine calls:

**WUInstall.exe /install_cache p:\temp**

and the update "Windows Malicious Software Removal Tool - August 2010 (KB890830)" is not yet present in the cache directory. First of all, the cache is checked for a directory with the same name as the Update ID (in this case 20110b11-6dd2-4153-a76d-e660ec78ebed), and the update is not found:

```
Checking cache for update: Windows Malicious Software Removal Tool - August
2010 (KB890830)
Update Windows Malicious Software Removal Tool - August 2010 (KB890830)
Multi-Lingual - Delta not found in cache directory p:\temp\20110b11-6dd2-
4153-a76d-e660ec78ebed
```

So, the update will be downloaded from the configured location (here, the Windows Update site) and copied into the cache folder:

```
Update bundle not found, downloading...
Copying downloads directly...
Downloading: http://download.windowsupdate.com/msdownload/update/software/
uprl/2010/08/windows-kb890830-v3.10-
delta_b6366a428af17c9c7c524ebd15aafe3608d3fa39.exe to p:\temp\20110b11-6dd2-
4153-a76d-e660ec78ebed/windows-kb890830-v3.10-
delta_b6366a428af17c9c7c524ebd15aafe3608d3fa39.exe ...
OK
Download OK
```

After the download is finished, WuInstall checks the cache again and finds the just downloaded update there and copies it to the local cache of the machine (the local cache is usually following directory: Windows \SoftwareDistribution\Download)

```
Checking cache again...
Update Windows Malicious Software Removal Tool - August 2010 (KB890830)
Multi-Lingual - Delta found in cache directory, copying to local cache... OK
```

After that, the download of the update is finished and it is installed.

Later, another machine (which also needs the KB890830 update) also calls:

**WUInstall.exe /install_cache p:\temp**

Now, the update is present in the cache directory and is copied from there without downloading it from the Internet or WSUS Server (which apparently saves a lot of Internet bandwidth since the cache directory is in the local network):

```
Checking cache for update: Windows Malicious Software Removal Tool - August
2010 (KB890830)
Update Windows Malicious Software Removal Tool - August 2010 (KB890830)
Multi-Lingual - Delta found in cache directory, copying to local cache...
OK
```

### /download_cache cache_dir

This option does the same as /install_cache, but only downloads the updates, without installing them (which can be used for example to populate the cache with updates, without installing them).

### /nocachedel

In combination with /install_cache or /download_cache: does not clear the cache directory on error.

### /clearlocalcache

#### Available in WuInstall 2.0

This option clears the local windows update cache.

The local cache is located at %systemroot%\SoftwareDistribution\Download.

## Criteria Options

**Contents**

### /criteria "query string"

Specifies the criteria with which the updates are searched on the server (Windows Update Site or WSUS).

If no criteria is specified, WuInstall searches by default for software (no drivers!) which has not yet been installed. So the default criteria is: IsInstalled=0 and Type='Driver'.

When an invalid criteria was specified, the search fails and WuInstall exits with code 4.

```
Invalid criteria !
```

When the criteria was valid, e.g. when you type

**WUInstall.exe /search /criteria "IsInstalled=1"**

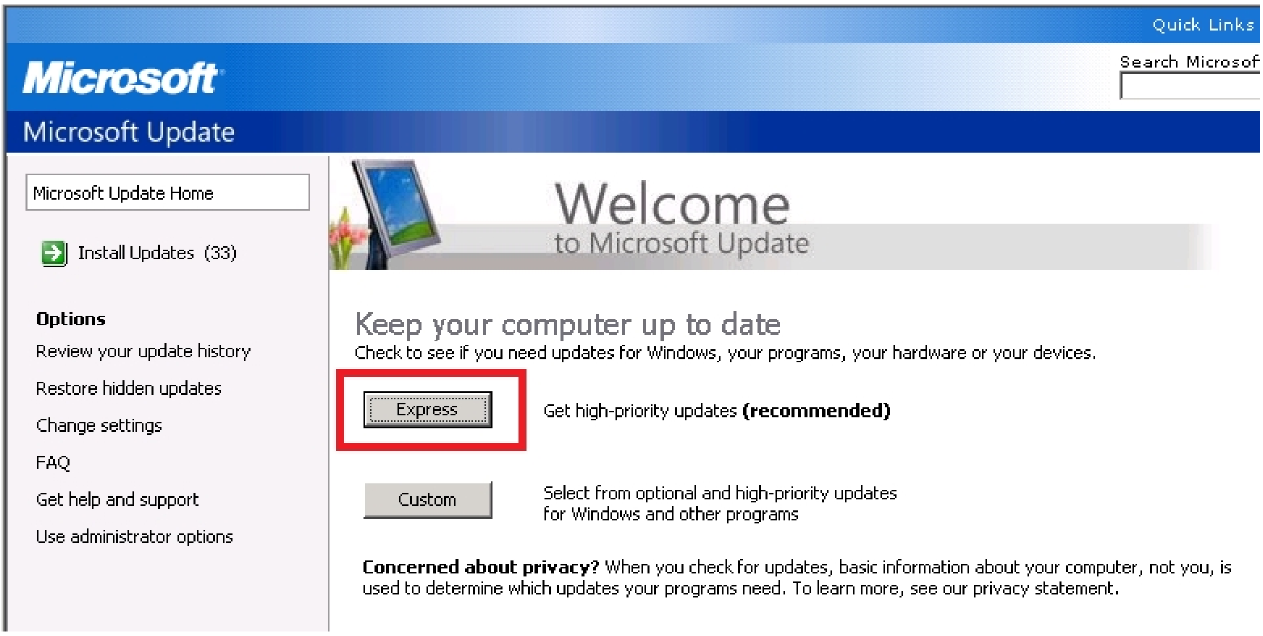
WuInstall displays

```
Searching for updates ... Criteria: IsInstalled=1
```

and includes only updates in the result list, which match this criteria.

For more details on criteria see http://msdn.microsoft.com/en-us/library/aa386526%28VS.85%29.aspx

To perform an express update like the express option on the windows update site



Keep your computer up to date

Check to see if you need updates for Windows, your programs, your hardware or your devices.

Get high-priority updates **(recommended)**

Select from optional and high-priority updates for Windows and other programs

**Concerned about privacy?** When you check for updates, basic information about your computer, not you, is used to determine which updates your programs need. To learn more, see our privacy statement.

use the command:

**WuInstall.exe /search /bypass_wsus /criteria "IsHidden=0 and IsInstalled=0 and IsAssigned=1"**

## /driveronly

Searches only for drivers which have not yet been installed. Equivalent to the criteria (IsInstalled=0 and Type='Driver').

## /includedriver

Searches for software and drivers which have not yet been installed. Equivalent to criteria (Type='Software' and IsInstalled=0) or (Type='Driver' and IsInstalled=0)

## /alltypes

Searches for all types of updates which have not yet been installed. Equivalent to criteria (IsInstalled=0).

# Filter Options

**Contents**

## /classification update_classification

Each update has a certain classification. It is possible to filter results of a search (which match the specified criteria) by classifications.

Classifications are:

- **Critical Updates:** Broadly released fixes for specific problems addressing critical, non-security related bugs.
- **Updates:** Broadly released fixes for specific problems addressing non-critical, non-security related bugs.
- **Update Rollups:** Cumulative set of hotfixes, security updates, critical updates, and updates packaged together for easy deployment. A rollup generally targets a specific area, such as security, or a specific component, such as Internet Information Services (IIS).
- **Security Updates:** Broadly released fixes for specific products, addressing security issues.
- **Service Packs:** Cumulative sets of all hotfixes, security updates, critical updates, and updates created since the release of the product. Service packs might also contain a limited number of customer-requested design changes or features.
- **Feature Packs:** New feature releases, usually rolled into products at the next release.
- **Definition Updates:** Updates to virus or other definition files.

You can combine classifications by just combining the letters:

C (Critical Updates)

U (Updates)

R (Update Rollups)

S (Security Updates)

E (Service Packs)

F (Feature Packs)

D (Definition Updates)

For example

**WUInstall.exe /search /criteria "IsInstalled=1" /classification CRSE** generates the following output:

```
Searching for updates ... Criteria: IsInstalled=1
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing those matching following classifications:
Critical Updates
Update Rollups
Security Updates
Service Packs
```

## /severity severity_level

Each security update has a certain severity level. It is possible to filter results of a search (which match the specified criteria) by severity levels.

Severity levels are:

- **Critical:** A security issue whose exploitation could allow the propagation of an Internet worm without user action.
- **Important:** A security issue whose exploitation could result in compromise of the confidentiality, integrity, or availability of users' data, or of the integrity or availability of processing resources.
- **Moderate:** Exploitation is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
- **Low:** A security issue whose exploitation is extremely difficult, or whose impact is minimal.
- **Unknown:** all updates, which are not a security update.

You can combine severity levels by just combining the letters:

C (Critical)

I (Important)

M (Moderate)

L (Low)

U (Unknown)

So the command

**WUInstall.exe /search /severity CIU**

filters the result list of the search to contain only updates with severity Critical, Important and Unknown:

```
Searching for updates ... Criteria: IsInstalled=0 and Type='Software'
Result Code: Succeeded
```

```
30 Updates found in total, filtering ...
Listing those matching following severities:
Critical
Important
Unknown
```

## /product "product_name"

Each update belongs to a product (for example Office 2007, Windows 7, Forefront Client Security, ...). It is also possible to filter results of a search (which match the specified criteria) by a product name.

No wildcards are possible, the product name has to be an exact match. Currently, you can only filter one product.

So for example the command

**WUInstall.exe /search /criteria IsInstalled=1 /product "Forefront Client Security"**

gives the following output:

```
Searching for updates ... Criteria: IsInstalled=1
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing product:  'Forefront Client Security'
```

## /noproduct "product_name"

This option is filters the search result and excludes the given product.

## /match "regexp"

This option filters each update in the result list if the title of an update matches the search string specified with the regexp parameter (in WuInstall Pro, also regular expressions are possible).

You can search, for example, just for all updates matching KB890830 with:

**WUInstall.exe /search /match KB890830 /criteria IsInstalled=1**

```
Searching for updates ... Criteria: IsInstalled=1
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing those matching search expression:  'KB890830'


1. Windows Malicious Software Removal Tool - August 2010 (KB890830)
 Severity: Unknown
 UpdateClassification - Update Rollups
 Product - Windows 7
```

With using regular expressions, you can do some more complex queries. For a good explanation of regular expressions see http://www.cs.tut.fi/~jkorpela/perl/regexp.html

For example, searching for all updates that match "Internet Explorer 8", the Term "Language" or "KB89", you type:

**WuInstall /search /match "(Language|KB93|Internet Explorer)"**

```
Searching for updates ... Criteria: IsInstalled=1
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing those matching search expression:  '(Language|KB89|Internet
Explorer)'


1. Update for Microsoft Office Language Pack 2007 (for Office Outlook 2007
with
Business Contact Manager ) (KB946307)
 Severity: Unknown
 UpdateClassification - Critical Updates
 Product - Office 2007

2. Update for Internet Explorer 8 Compatibility View List for Windows 7
(KB98263
2)
 Severity: Unknown
 UpdateClassification - Updates
 Product - Windows 7

3. Update for Internet Explorer 8 Compatibility View List for Windows 7
(KB98266
4)
 Severity: Unknown
 UpdateClassification - Updates
 Product - Windows 7

4. Cumulative Security Update for Internet Explorer 8 for Windows 7
(KB2183461)

 Severity: Critical
 UpdateClassification - Security Updates
 Product - Windows 7

5. Windows Malicious Software Removal Tool - August 2010 (KB890830)
 Severity: Unknown
 UpdateClassification - Update Rollups
 Product - Windows 7
```

## /nomatch "regex"

This is quite the opposite of the /match option: it filters out all the updates that do match the regular expression and displays the rest (=the updates which do NOT match).

Obviously it is very useful for excluding certain updates.

For example, to filter out all updates that do not contain the words "Microsoft" or "Windows" (there are not many updates that do so ...) use:

**WUInstall.exe /search /criteria "IsInstalled=1" /nomatch (Windows|Microsoft)**

```
Searching for updates ... Criteria: IsInstalled=1
```

```
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing those NOT matching search expression:  '(Windows|Microsoft)'


1. MSXML 6.0 RTM Security Update  (925673)
 Severity: Critical
 UpdateClassification - Security Updates
 Product - SQL Server Feature Pack

2. Update for Business Contact Manager for Outlook 2007 (KB946140)
 Severity: Unknown
 UpdateClassification - Critical Updates
 Product - Office 2007

3. Security Update for SQL Server 2005 Service Pack 3 (KB970892)
 Severity: Critical
 UpdateClassification - Security Updates
 Product - SQL Server 2005

4. Update Rollup 10 for Exchange Server 2007 Service Pack 1 (KB981407)
 Severity: Unknown
 Product - Exchange Server 2007
 UpdateClassification - Update Rollups
```

Of course, /match and /nomatch can also be combined.

## /matchfile file

With the /matchfile option you can specify multiple /match parameters.

Just write the desired update matches (regular expressions are supported) into a file and specify this file as the matchfile.

For example, create the file "updateList.txt" with the following content:

```
Security State Assessment 1.0
KB963671
KB975467
.NET Framework 3.5 SP1 Security
```

The match entries are separated with a newline. The file has to be either in UTF-8, ANSI or DOS character set. UTF-16 is not supported.

WuInstall lists all the updates in the result list, which match AT LEAST ONE of the lines in the file.

If you type

**WUInstall.exe /search /matchfile c:\updateList.txt**

WuInstall gives for example the following result list:

```
95 Updates found in total, filtering ...
Listing those matching search expressions in file:  'c:\updateList.txt'
```

```
1. Update for Microsoft Script Editor Help (KB963671)
 Severity: Unknown
 Product - Office 2007
 UpdateClassification - Updates

2. Security Update for Windows 7 (KB975467)
 Severity: Important
 UpdateClassification - Security Updates
 Product - Windows 7

3. Definition Update for Microsoft Forefront Client Security (Security State
Assessment 1.0.1710.103 Full)
 Severity: Unknown
 UpdateClassification - Definition Updates
 Product - Forefront Client Security

4. Microsoft .NET Framework 3.5 SP1 Security Update for Windows 7 x86
(KB979916)
 Severity: Important
 UpdateClassification - Security Updates
 Product - Windows 7
```

This option is useful when you keep a central list of updates to be rolled out on your machines in a shared directory (use this list as the matchfile).

## /nomatchfile file

/nomatchfile is quite the opposite of the /matchfile option: use it if you want to specify multiple /nomatch paramters (if you want to exclude a list of updates).

WuInstall lists all the updates in the result list, which match NONE of the lines in the file.

Just write the update matches that should be excluded (regular expressions are supported) into a file and specify this file as the nomatchfile.

For example, create the file "exclude.txt" with the following content:

```
Windows
Microsoft
925673
Office
```

The exclude entries are separated with a newline. The file has to be either in UTF-8, ANSI or DOS character set. UTF-16 is not supported.

If you type

**WUInstall.exe /search /nomatchfile c:\exclude.txt**

```
WuInstall puts out for example the following result list:

95 Updates found in total, filtering ...
Listing those NOT matching search expressions in file:  'c:\exclude.txt'
```

```
1. Update for Business Contact Manager for Outlook 2007 (KB946140)
 Severity: Unknown
 UpdateClassification - Critical Updates
 Product - Office 2007

2. Security Update for SQL Server 2005 Service Pack 3 (KB970892)
 Severity: Critical
 UpdateClassification - Security Updates
 Product - SQL Server 2005

3. Update Rollup 10 for Exchange Server 2007 Service Pack 1 (KB981407)
 Severity: Unknown
 Product - Exchange Server 2007
 UpdateClassification - Update Rollups
```

## /isdownloaded 1|0

This options filters updates which have already been downloaded to the local cache.

You can call for example WuInstall with the /download options combined with some filters to download all available updates:

```
30 Updates found in total, filtering ...
Listing those matching search expression:  'Greek'


1. Greek Language Pack - Windows 7 (KB972813)
 Severity: Unknown
 UpdateClassification - Updates
 Product - Windows 7 Language Packs

Downloading updates ...
Downloading Greek Language Pack - Windows 7 (KB972813):
```

and then later use:

**WUInstall.exe /install /isdownloaded 1**

You will get following output:

```
30 Updates found in total, filtering ...
Listing only updates which are already downloaded


1. Greek Language Pack - Windows 7 (KB972813)
 Severity: Unknown
 UpdateClassification - Updates
 Product - Windows 7 Language Packs
```

Deleting updates manually from the local download cache (usually Windows\SoftwareDistribution\Download) might cause this option to work incorrectly, because even deleted they are still considered as downloaded.

## /releasedate [lt|eq|gt]:[YYYY-MM-DD] or [-#Days]

### Available in WuInstall 2.0

This option filters updates by their release date and can be used in two different ways:

First with lt, eq or gt followed by the date you will filter.

- lt: updates where the release date is before the given date
- eq: updates where the release date is equals to the given date
- gt: updates where the release date is after the given date

Example:

**WUInstall.exe /search /releasedate lt:2014-05-06**

You will get following output:

```
3 Updates found in total, filtering ...
Listing those where release date lt 2014-05-06


1. Microsoft Visual Studio 2010 Service Pack 1
 Severity: Unknown
 UpdateClassification - Service Packs
 Product - Visual Studio 2010
 ID: 230b82d1-3abd-471a-a4f9-23f97fb857d9
 Releasedate: 2012-04-05
 Eula Accepted: true
```

The second way to use this options is to specify a number of days. This only lists all updates with release date after today minus the given days.

Example:

**WUInstall.exe /search /releasedate -10**

This will list all updates which are not older than 10 days:

```
3 Updates found in total, filtering ...
Listing those with release date after 2014-04-26


1. Definition Update for Windows Defender - KB2267602 (Definition
1.173.1428.0)
 Severity: Unknown
 UpdateClassification - Definition Updates
 Product - Windows Defender
 ID: 4f25e1f8-186e-4d0b-b18c-6dab7c081c09
 Releasedate: 2014-05-06
 Eula Accepted: true
```

## /maxUpdateCount number

This update specifies the maximal number of updates which are handled by a WuInstall call.

If there are more updates available, only the first updates up to the given number are handled.

# WSUS Options

**Contents**

## /bypass_wsus !!!DEPRECATED!!!

This option is deprecated, use /UseUpdateService instead.

The bypass_wsus option bypasses any configured WSUS and forces WuInstall to access the Windows Update site.

If a WSUS is configured, WuInstall changes the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU, Value UseWuServer from 1 to 0, which means that no WSUS should be used.

If you type

**WUInstall.exe /search /bypass_wsus**

and a WSUS is configured, the output is:

```
You are using WSUS Server http://mywsus.local:8530
bypass_wsus specified, bypassing WSUS ... successful!
```

After the execution of WuInstall, the initial state is restored by resetting the UseWuServer value to 1.

```
bypass_wsus specified, reactivating WSUS ... successful!
```

When the execution of WuInstall is aborted for some reason before WuInstall can reset the registry value, it have to be reset manually!

## /use_wsus wsus_address

The /use_wsus option works similar to the /bypass_wsus option, but you can specifiy another WSUS which is valid during the execution of WuInstall (or it allows you to switch from no WSUS at all to a certain WSUS) .

WuInstall manipulates the following registry keys:

- HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU\UseWuServer: is set to 1
- HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\WUServer: is set to the specified wsus_address
- HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\WUStatusServer: is set to the specified wsus_address

and stores the initial values for resetting them after WuInstall has finished.

For example, if you have WSUS http://mywsus.local:8530 configured, and want to go to the WSUS http://mywsus2.local:8530, you type:

**WUInstall.exe /search /use_wsus http://mywsus2.local:8530**

and WuInstall puts out:

```
You are using WSUS Server http://mywsus.local:8530
use_wsus specified ....  WUServer and WUStatusServer successfully set to
http://mywsus2.local:8530
```

After the execution of WuInstall, the initial state is restored (in this example, the WSUS is reset to http://mywsus.local:8530)

```
use_wsus specified, deactivating WSUS again
resetting  WUServer to http://mywsus.local:8530, WUStatusServer to http://mywsus.local:8530 ...
successful!
```

When the execution of WuInstall is aborted for some reason before WuInstall can reset the registry values, they have to be reset manually!

## /use_mu_on_error

When an error occurs trying to install an update using WSUS, WuInstall automatically tries again to install the updates bypassing the WSUS.

# Proxy Options

### Contents

## /disable_winhttp_proxy

Disables the winhttp Proxy. Therefore WuInstall deletes the registry key HKEY_LOCAL_MACHINE \SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Connections. After execution WUInstall enableds the Proxy again.

## /disable_internet_proxy

This option disables the internet proxy, after executing WUInstall the proxy is enabled again.

## /proxyauto

### Available in WuInstall 2.0

Îf this option is set, the web proxy settings will be detected automatically.

### /proxyaddress "address:port"

#### Available in WuInstall 2.0

Sets the address and the decimal port number of the proxy.

### /proxyuser "username"

#### Available in WuInstall 2.0

Sets the user name to submit to the proxy server for authentication.

### /proxypass "password"

#### Available in WuInstall 2.0

Sets the password to submit to the proxy server for authentication.

## Other Options

**Contents**

### /autoaccepteula

For some updates, an EULA (End-User License Agreement) has to be accepted by the user. With the option /autoaccepteula, all EULAs are accepted automatically before installing.

So the update installation works without any user interaction, the process can be completely unattended.

This works only when WuInstall is called by an Administrator or a power user account.

## /logfile file-to-log

This options logs the console output of WuInstall to the specified logfile (each output line also contains the current timestamp).

For example, to log all already installed downloads:

**WUInstall.exe /search /criteria "IsInstalled=1" /logfile "c:\data\WuInstallLog.txt"**

WuInstallLog.txt contains for example the following lines:

```
2010/08/24-08:10:37   Searching for updates ... Criteria: IsInstalled=1
2010/08/24-08:10:40
2010/08/24-08:10:40 Result Code: Succeeded
2010/08/24-08:10:40   50 Updates found in total, filtering ...
2010/08/24-08:10:40   1. German Language Pack - Windows 7 (KB972813)
Severity: Unknown
2010/08/24-08:10:40   UpdateClassification - Updates  Product - Windows 7
Language Packs
2010/08/24-08:10:40   2. Windows Server Update Services 3.0 SP2 (KB972455)
Severity: Unknown
2010/08/24-08:10:40   UpdateClassification - Service Packs  Product - Windows
7
2010/08/24-08:10:40   3. French Language Pack - Windows 7 (KB972813)
Severity: Unknown
2010/08/24-08:10:40   UpdateClassification - Updates  Product - Windows 7
Language Packs
```

This option is obviously very useful when testing and debugging WuInstall scripts or when running WuInstall with a scheduled job or on remote machines.

## /logfile_append file-to-log

The option /logfile_append is similar to /logfile with the difference that the entries are appended at the end of the specified file, instead of creating a new file.

## /log_as_ansi

### Available in WuInstall 2.0

This options creates a ANSI-logfile. Must be combined with /logfile or /log_append.

.

## /maxruntime minutes

This option starts a timer with the specified number of minutes at the start of WuInstall and exits WuInstall (if it is not finished until the timer expires) with exit code 12. It exits at any case, no matter how far the execution WuInstall is.

For example, with /maxruntime 5, WuInstall exits after 5 minutes displaying

```
Maxruntime timeout elapsed: 5, exiting process!
```

This option is intended for machines, where downloads take a lot of time or machines which hang for some reason, to avoid WuInstall from running endlessly long.

Use this option with caution, especially when using the /use_wsus or /bypass_wsus options!

## /simple_error

With this option, WuInstall returns 0 for any success and 1 for any error, regardless of the reason.

Which means, if the return code would be 0, 2, 3, 5 or 10, WuInstall returns 0 and displays the message

```
Simple result specified, returning 0 for success!
```

In any other case (the remaining codes 1, 4, 6, 11, 12), WuInstall returns 1 and displays the message

```
Simple result specified, returning 1 for error!
```

## /show_progress

This option shows the progress for the download of every single update and the installation of the updates, using asynchronous downloads / installation.

On some machines under certain circumstances, the /show_progress option might cause problems, so use it only when manually testing scripts or searching for problems on a single workstation. It is not advised to use it in automated scripts on different workstations.

This feature is not recommended for the productive use of WuInstall!

## /offline

This options makes WuInstall NOT connecting to the WSUS Server or Microsoft Update site when searching updates, instead it searches offline.

This option is quite useful for searching already installed updates. The criteria "IsInstalled=1" and the /uninstall option automatically trigger an offline search.

For example, check if the Update KB982214 is installed on a machine:

**WUInstall.exe /search /criteria "IsInstalled=1" /offline /match "KB982214"**

```
/offline specified, searching offline...

Searching for updates ... Criteria: IsInstalled=1
Result Code: Succeeded

96 Updates found in total, filtering ...
Listing those matching search expression:  'KB982214'


1. Security Update for Windows 7 (KB982214)
 Severity: Important
 UpdateClassification - Security Updates
 Product - Windows 7
```

When not using IsInstalled=1, the search result list usually contains the updates found at the last online search.

## /quiet

The update-installer is forced to do an installation without user interaction.

## /disableprompt

Disable showing source prompts to a user when installing the updates.

## /silent

Suppresses any console output (logfile option is still working).

## /donothing [retcode]

WuInstall takes no action and returns the specified retcode, 0 by default.

## /xmlout file

Writes an XML report to the specified file. This file contains additional information about the machine.

```
<?xml version="1.0" encoding="UTF-8"?>
<wuinstall runtime_seconds="10" runtime="0d 00:00:10">
      <system-info>
            <wuinstall-version>WuInstall Version 2.0 Pro (codename AE), Build
2014-02-03</wuinstall-version>
            <windows>
                  <version>Microsoft Windows [Version 6.2.9200]</version>
                  <service-pack>No Service Pack installed!</service-pack>
                  <windows-update-agent>
                        <major-version>8</major-version>
                        <minor-version>0</minor-version>
                        <product-version>7.9.9600.16384</product-version>
                  </windows-update-agent>
                  <domain>WORKGROUP</domain>
                  <username>Michael</username>
                  <computername>MICHAEL-PC</computername>
            </windows>
            <processor-architecture>x86</processor-architecture>
            <WOW64>true</WOW64>
      </system-info>
      <wsus></wsus>
      <parameters>/search /xmlout c:\tmp\wuinstall.xml</parameters>
      <updates count="1">
      <update>
            <title>Definition Update for Windows Defender - KB2267602
(Definition 1.173.1428.0)</title>
            <description>Install this update to revise the definition files
that are used to detect viruses, spyware, and other potentially unwanted
software. Once you have installed this item, it cannot be removed.</
description>
            <download_priority>Normal</download_priority>
            <eula_accepted>true</eula_accepted>
            <identity>4f25e1f8-186e-4d0b-b18c-6dab7c081c09</identity>
            <is_beta>false</is_beta>
            <is_downloaded>false</is_downloaded>
            <is_hidden>false</is_hidden>
            <is_installed>false</is_installed>
            <is_mandatory>false</is_mandatory>
            <is_present>true</is_present>
            <is_uninstallable>false</is_uninstallable>
            <kb_article_ids>
```

```
                  <kb_article_id>2267602</kb_article_id>
            </kb_article_ids>
            <last_deployment_change_time>2014-05-06</
last_deployment_change_time>
            <max_download_size>127745424</max_download_size>
            <min_download_size>0</min_download_size>
            <msrc_severity></msrc_severity>
            <recommended_cpu_speed>0</recommended_cpu_speed>
            <recommended_hard_disk_space>0</recommended_hard_disk_space>
            <recommended_memory>0</recommended_memory>
            <support_url>http://go.microsoft.com/fwlink/?LinkId=52661</
support_url>
            <update_type>Software</update_type>
            <categories>
            <category>
            <id>e0789628-ce08-4437-be74-2495b842f43b</id>
            <description>A broadly-released and frequent software update
containing additions to a product's definition database. Definition databases
are often used to detect objects with specific attributes, such as malicious
code, phishing Web sites, or junk e-mail.</description>
            <name>Definition Updates</name>
            <order>-1</order>
            <type>UpdateClassification</type>
            </category>
            <category>
            <id>8c3fcc84-7410-4a95-8b89-a166a0190486</id>
            <description>Windows Defender helps protect your computer against
pop-ups, slow performance, and security threats caused by spyware and other
unwanted software.</description>
            <name>Windows Defender</name>
            <order>-1</order>
            <type>Product</type>
            </category>
            </categories>
            <download_result_code>NotStarted</download_result_code>
            <download_hresult></download_hresult>
            <download_hresult_text></download_hresult_text>
            <installation_result_code>NotStarted</installation_result_code>
            <installation_hresult></installation_hresult>
            <installation_hresult_text></installation_hresult_text>
            <reboot_required>false</reboot_required>
      </update>
      </updates>
      <result>
            <search_result>Succeeded</search_result>
            <return_code>0</return_code>
            <return_text></return_text>
            <reboot_required>false</reboot_required>
      </result>
</wuinstall>
```

## /includeRecommendedUpdates

Includes recommended and optional updates.

## /disable_ie_firstruncustomize

Disables the internet explorer 8 set up dialog.

## /targetgroup "targetgroup"

### Available in WuInstall 2.0

One of the nice things about WSUS is that it allows you to use client side targeting. The idea behind client side targeting is that you can set up different computer groups, and you can roll out updates on a group basis. Client side targeting isn't used by default, but if you decide to use it, you can use this option to assign the computer to a targetgroup.

## /offlinescan "scanfile"

### Available in WuInstall 2.0

This option can be used to search for updates against a cab-file. You can download the cab-file from the following link:
http://go.microsoft.com/fwlink/?LinkID=74689
The .cab file must be on a local directory, a UNC path won't work. We suggest to copy the .cab file into a local directory before executing WuInstall.

This can be used to update computers which have no access to a WSUS or to the internet. Therefore you have to combine it with the /install_cache option. First all updates have to be downloaded to the cache directory.

You have to execute WUInstall as administrator to use this option, otherwise it won't work.

## /options [file://]|[http://]path

### Available in WuInstall 2.0

This options allowes it to specify the options in a file which is either on the disk (use file://) or available on the internet (use http://).

In the options-file each option have to be in a seperate line.

## /tempdir "path"

### Available in WuInstall 2.0

Use this option to set the temp directory to the given path. By default the temp directory is set to %temp%.

## /force

This option forces the update installation.

## /remote "paexec_options"

### Available in WuInstall 2.0

This option executes WuInstall on a remote machine using PAExec. The hostename an login credentials have to be specified in the argument ("\\hostename -u user -p password"). All other WuInstall options can be specified as usual.

Example:
**WUInstall.exe /remote "\\testhost -u myuser -p mypasswd" /search**

This command executes WuInstall on the host named "testhost" and performs a search.

For more information about PAExec visit http://www.poweradmin.com/paexec/.

## /third_party_progress tag

### Available in WuInstall 2.0

This option adds additional output, each line will start and end with the given tag. This makes it easier to parse the output.
Following lines will be added:

After searching and applying filters: `%d updates matching filter` - %d represents the number of matching updates
Before each download: `starting download %d/%d, (%s)` - %d/%d for example starting download of the first of three (1/3) updates, (%s) the uuid of the current update
After each download: `finished download %d/%d, (%s)` - %d/%d for example finished download of the first of three (1/3) updates, (%s) the uuid of the current update
Before each installation: `starting install %d/%d, (%s)` - %d/%d for example starting install of the first of three (1/3) updates, (%s) the uuid of the current update
After each installation: `finished install %d/%d, (%s) result=%s` - %d/%d for example finished install of the first of three (1/3) updates, (%s) the uuid of the current update, result=%s result status for the current installation
Last line of output: `return code: %d` - %d overall return code from WuInstall

For the installation output the /show_progress have to be used additionally.

## /refresh_last_update_timestamp

### Available in WuInstall 2.0

This option refreshes the timestamps for the last successful search/install, which will be displayed in the Windows Update center.

## /UseUpdateService MicrosoftUpdate|WindowsUpdate|WSUS|Default

This options sets the specified service which will be used for searching for updates.

# WuInstall Return Codes

### Contents

## Return Codes

- 0: successful, no reboot required
- 1: at least one error occurred, no reboot required
- 2: no more updates available
- 3: no updates available that match your search (in combination with /severity, /classification, /product, /criteria, /match, /nomatch, /matchfile, /nomatchfile)
- 4: invalid criteria specified (in combination with /criteria)
- 5: reboot/shutdown initialized successful (in combination with /reboot or /shutdown)
- 6: reboot/shutdown failed (in combination with /reboot or /shutdown)
- 7: syntaxerror, wrong command
- 8: invalid version, expiration date reached
- 10: successful, reboot required
- 11: at least one error occurred, reboot required
- 12: timeout reached (in combination with /maxtruntime)#
- 15: Reebootcycle successfull
- 16: Reebootcycle failed

## Error Codes

| Error Code | Error String | Description |
|---|---|---|
| 0x80240044 | WU_E_PER_MACHINE_UPDATE_ACCESS_DENIED | Only administrators can perform this operation on per-machine updates. |
| 0x80248013 | WU_E_DS_DUPLICATEUPDATEID | The server sent the same update to the client with two different revision IDs. wuinstall.exe exited on <MyServerName> with error code 1. |
| 0x80244021 | WU_E_PT_HTTP_STATUS_BAD_GATEWAY | Same as HTTP status 502 The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request. |
| 0x80072EE6 | ERROR_INTERNET_UNRECOGNIZED_SCHEME | The URL scheme could not be recognized or is not supported. |

# HowTo

## 1. Basic functions

There are three basic switches. Exactly one of them must be specified in order for wuInstall to do something. It makes no sense to specify more than one of those switches (e.g. wuInstall /search /install), because download automatically includes a search and install automatically includes search and download.

### 1.1. Searching for updates

The simplest command in WuInstall is just searching for updates:

```
WuInstall /search
```

It lists all update which are available. It either searches on the Windows Update Server in the Internet, or, if configured, on your WSUS. It has no impact on your system at all, it just does a search ans lists what was found.

### 1.2. Downloading updates

You can download updates with WuInstall with

```
WuInstall /download
```

It does the same search as WuInstall /search and then downloads the updates which were found. WuInstall itself does not download anything, but it triggers the Microsoft Windows Update Engine, which downloads the updates.

### 1.2.1. Downloading updates to a certain directory

The download option can be extended, in order to also copy the installation packages included in the download to a certain directory (e.g. for manual installation and distribution)

```
WuInstall /download_to "c:\updates_download"
```

Does exactly the same as WuInstall /download, but afterwards it copies the contents of the downloaded updates (usually .exe or .msi-files) for example to c:\updates_download, from where they can be installed manually or copied and used for patching other machines

### 1.3. Installing updates

To install the updates, you use:

```
wuInstall /install
```

It again searches the updates, downloads them (if they have not already been downloaded by a previous wuInstall /download command) and then installs them.

## 2. Selecting updates

With various switches, you can narrow the results of the search for updates and therefore also the updates which are downloaded and installed. You can use criteria queries, search updates which match certain names, severities, products or classifications.

**Tip:** In subsequent examples, we only use the /search option to illustrate how to select updates. This option is ideal to test certain switches, because it just lists the updates and has no further impact on the system. Of course, also the /download or /install switch can be used with all following the examples.

### 2.1. Specifying criterias

With /criteria you can specifiy a query string, in order to change the search. The default criteria (when no /criteria is specified) is "IsInstalled=0 and Type='Software'".

**Tip:** For how the query string can be built see MSDN

You can, for example search for updates which are already installed:

```
wuInstall /search /criteria "IsInstalled=1"
```

(here, obviously, a /download or /install makes no sense)

Furthermore you can also just select Drivers:

```
wuInstall /search /criteria "IsInstalled=0 and Type='Driver'"
```

or Drivers and Software:

```
wuInstall /search /alltypes
```

which is equivalent to:

```
wuInstall /search /criteria "IsInstalled=0"
```

### 2.2. Matching specific search strings

With the /match option, you can get updates of which the title matches a specific search string.

**Tip:** You can test the /match option for example with /criteria "IsInstalled=1", because then you (usually) have a very long list of updates in your search.

For example, you can search for a specific update:

```
wuInstall /search /match "KB956844"
```

only lists the "*Security Update for Windows XP (KB956844)*"
In the WuInstall Pro Version /match also supports regular expressions, for example find all updates which have the terms “Language” OR “KB93” OR “Live” in their title:

```
WuInstall /search /match "(Language|KB93|Live)"
```

### 2.3. Not Matching specific search strings

In the WuInstall Pro Version, there is, corresponding to the /match-switch also a /nomatch-switch For example, search all Updates which do NOT match "SQL" OR "Windows" OR "KB"

```
WuInstall /search /nomatch "(SQL|Windows|KB)"
```

/match and /nomatch can, of course, also be used in combination, for example search all Updates which match "SQL" OR "Office" but do not Match "KB" or "Microsoft"

```
WuInstall /search /match "(SQL|Office)" /nomatch "(KB|Microsoft)"
```

### 2.4. Search string in files

It is also possible to put search strings in a file. For example, you create a file match.txt with the following search string (seperated by a newline):
c:\Data\match.txt

```
KB93
KB94
KB95
Office
Internet Explorer 8
```

If you execute

```
wuInstall /search /matchfile c:\Data\match.txt
```

Your will get all updates matching KB93, KB94, KB95, Office or Internet Explorer 8.
There is also a /nomatchfile switch which does the opposite, which excludes updates matching one of the search terms in the file.

```
wuInstall /search /nomatchfile c:\Data\match.txt
```

Regular expressions are also possible as search expressions in the files.

### 2.5. Severity

Each security update has a certain severity. There are five possible severities:

- Critical (C): A security issue whose exploitation could allow the propagation of an Internet worm without user action.
- Important (I): A security issue whose exploitation could result in compromise of the confidentiality, integrity, or availability of users' data, or of the integrity or availability of processing resources.
- Moderate (M): Exploitation is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.

- Low (L): A security issue whose exploitation is extremely difficult, or whose impact is minimal.
- Unknown (U): No information about the severity is available.

In letter the brackets is the code for the /severity - option. Just put the letters together to select multiple severities.
For example, all critical and important updates:

```
WuInstall /search /severity CI
```

**Tip:** The severity is only filled for security updates, so it is not really suitable for finding high priority updates as security updates usually ARE high prioritoy updates. Classification is better for that purpose.

### 2.6. Classification

Each update belongs to one of those classifications:

- Critical Updates (C)
- Security Updates (S)
- Updates (U)
- Update Rollups (R)
- Services Packs (E)
- Feature Packs (F)
- Definition Updates (D)

Just put the letters in the bracket together to select multiple severities. For example, select all security updates, critical updates and Updates:

```
WuInstall /search /classification SCU
```

**Tip:** High priority updates can best be found with the classification CRS (Critical, Security and Update Rollups), however sometimes also U (Updates) can be high priority. Generally speaking, selecting which updates are high priority and which are not is up to the administrator, there is no general rule how to automatically match the high priority categorization on the microsoft update website

### 2.7. Product

Each update belongs to a Microsoft product, for example Office 2003, Windows XP, Visual Studio 2008, Silverlight, ...
The /product switch enables to search by one product.
For example, find all Updates for Visual Studio 2008:

**Tip:** The product name has to match exactly, so just specifying "Visual Studio" here would produce no results.

```
WuInstall /search /product "Visual Studio 2008"
```

**Tip:** All update selection options (/match /nomatch /matchfile /nomatchfile /severity /classification /product) can be used in combination with each other!

## 3. Additional options

There are some additional options with which you can control the behavior WuInstall and the installation of updates. For example rebooting the system, bypassing the wsus or accepting EULAs without user interaction.

### 3.1. Reboot

The reboot-option just reboots the system n seconds after WuInstall has finished. It can be used together with search, download or install and it can also be used standalone.
For example, reboot in 30 seconds:

```
WuInstall /reboot 30
```

If no time interval is specified, 10 seconds is the default delay.
Reboot 10 seconds after the updates have been installed:

```
WuInstall /install /reboot
```

**Tip:** The /reboot option reboots the system after executing WuInstall unconditionally after the specified number of seconds. The result of WuInstall has no influence whether if there is a reboot or not.

### 3.2. Rebootcycle

With this option complete cumulative updating of windows systems becomes possible. That means installing updates, rebooting and installing further updates will be automated. After performing an update and rebooting afterwards, WuInstall starts again before the Windows Logon and executes the same command again, which was done before the reboot. The /rebootcycle option uses the Windows Task Scheduler to restart WuInstall.

**Tip:** You can limit this rebootcycle option to a maximum number of reboots to prevent of getting stuck in an endless loop. Default is 3.

The /rebootcycle option works together with /reboot or /reboot_if_needed.

When the /rebootcycle is used WuInstall outputs in the console:

```
/rebootcycle is specified, going to run this wuinstall command at most x
times again at startup if reboot is triggered
```

Example usage:

```
wuInstall /install /logfile_append c:\loginst.txt /reboot_if_needed /
rebootcycle 2
```

This command installs all available software updates. If necessary the system will reboot after an update installation and WuInstall will execute the command until all pending available updates are installed. In this case the numbers of reboots is limited to a maximum of 2 in any case. Also the /logfile_append option is set to have a report about the whole installation process including all reboots.

### 3.3. Rebootcycles and shutdown when finished

As already mentioned the /rebootcycle option automates reboots inbetween of updates. In many cases it might be necessary to shutdown the machine after having all updates done. For example if you want to schedule your windows updates during the night.

Example usage, which installs all available updates and shuts down the system when finished:

```
WuInstall /install /rebootcycle /shutdown
```

### 3.4. Bypassing the WSUS

If you have a WSUS, but want to go for the Windows Update Site instead of WSUS for some reason, you can use the option /bypass_wsus to bypass the WSUS server for the time WuInstall is executed.
For example, bypass wsus and install update KB956844 from the Windows Update Site:

```
WuInstall /install /bypass_wsus /match "KB956844"
```

After execution of WuInstall, WSUS is re-activated for your system. This option automatically executes the Sc stop wuauserv and Sc start wuauserv commands in order for the bypass to take effect.

**Tip:** If WuInstall fails for some reason or is aborted before it finishes, the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU, Value UseWuServer is still set to 0. You have to set it to 1 manually, in order to tell your system to use WSUS again.

### 3.5. Selecting another WSUS

Quite the opposite of the bypass_wsus option is the /use_wsus option. If you want to choose another WSUS or if you have no WSUS configured at the machine and want to get updates from a certain WSUS, the use_wsus option can force WuInstall to use a certain WSUS
For example, install your updates from the server wsus2 instead of wsus1:

```
WuInstall /install /use_wsus http://wsus2
```

WuInstall writes something like this:

```
You are using WSUS Server http://wsus1
use_wsus specified ....  WUServer and WUStatusServer successfully set to
http://wsus2
```

After execution of WuInstall, the initial state is restored (e.g. wsus1 is re-activated or WSUS is switched off again if it was off)

```
use_wsus specified, reactivating  WUServer http://wsus1,  WUStatusServer
http://wsus1 ... successful!
```

**Tip:** If WuInstall fails for some reason or is aborted before it finishes, the registry keys HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU\UseWuServer, HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\WUServer, HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\WUStatusServer have to be reset manually to the initial state

### 3.6. Accepting EULAs

Some updates require the user to accept an EULA in order to install. To automate this, the /autoaccepteula option automatically accepts all EULAs for updates which require this, without any user interaction or interruption.
Example:

```
WuInstall /install /autoaccepteula
```

# FAQ

## 1. Technical

### How to use WuInstall?

WuInstall is a command line tool, so you have to run "cmd" to get to the command line. If you double click on WuInstall, you only get a window showing a short description of the parameters. We recommend to run WuInstall as administrator to avoid troubles with access rights.

Basically, you can do three things with WuInstall:

- Searching for Updates: WuInstall /search
  → searches for all available updates, either (if configured) on the internal WSUS or in the internet on the Microsoft Update Server and gives a list of the updates found
- Downloading updates: WuInstall /download
  → does a search like in a), and downloads the updates found
- Installing updates: WuInstall /install
  → does a search first, then it downloads and finally installs the updates found

There are several options for selecting not all, but only specific updates: /severity, /match, /nomatch, /matchfile, /nomatchfile, /classification, /product, /criteria

Beyond that, there are some other very useful options:

/bypass_wsus: switches the WSUS off while executing WuInstall, to force going for the Microsoft Update Server in the internet
/autoaccapteula: automatically accepts EULAs for all updates which have to be installed → no user interaction required
/reboot [nseconds]: after executing WuInstall, the system is rebooted after n seconds

For more infos, see also the WuInstall HowTos

## How to get "Important" Updates only?

To get the same updates as you get when selected "Important" Updates on the windows update page, please try the following criteria option:

```
wuinstall /search /criteria "IsHidden=0 and IsInstalled=0 and IsAssigned=1"
```

## How do I call WuInstall on a remote machine using psexec?

Use the following command:

```
psexec \\server -c -s
-u domain\administrator -p password \\share\path_to_wuinstall\WUInstall.exe /
search
```

\\server → the remote Windows machine
\\share\path_to_wuinstall\ → path to WuInstall executable
-u domain\administrator → local admin oder domain admin user on the machine \\server
-p password → password of the user
-c → copies WuInstall on the remote system
-s → executes the command (wuinstall.exe) on the local system

## How to use WuInstall with a proxy server?

- When using a proxy which needs authentifcation, the user who executes wuinstall has to have the right to access the internet via the proxy.
- When using a WSUS via proxy, the wsus should be in the proxy exception list

## Does windows reboot automatically after WuInstall installed updates which need to reboot the machine?

No!

But WuInstall returns a status code, which indicates whether a reboot would be necessary or not.

There is also the /reboot-option in WuInstall (it can be used in combination with all other options). It reboots windows, after executing WuInstall, at any case.
`WuInstall /reboot` just reboots the system, without doing anything else.

## How does the /criteria option work?

For a detailed description of the query strings see MSDN

Some common examples:

- `WuInstall /search /criteria "IsInstalled=0 and Type='Software'"`: searches updates of type software, which are not yet installed (default-setting when the /criteria option is not specified)

- `WuInstall /search /criteria "IsInstalled=1"`: searches all updates, which are already installed on the machine (here, the options /install or /download are obviously not applicable, they will have no effect)

## Why are Updates marked as hidden also installed?

The default criteria option (see above) includes hidden and non-hidden updates. If you add “IsHidden=0” then the updates you marked as hidden will be skipped. An example would be: WuInstall /install /criteria "IsInstalled=0 and Type='Software' and IsHidden=0"

## Where does WuInstall download the updates?

The file location where WuInstall downloads the updates depends on your Windows Version and System Configuration. It is the same directory the "ordinary" Windows Update uses as well.
Usually, this directory is *Windows\SoftwareDistribution\Download*

## Does WuInstall work with Windows 7?

Yes, WuInstall has been tested with Windows 7, 32-bit and 64-bit, and it works just as well as with all other Windows versions. We recommend to run WuInstall as administrator to avoid troubles with access rights.

## How to search for special Updates or how to use Regular Expressions for /match / nomatch?

If you want to select special Updates by using regular expressions (e.g for the switches /match or / nomatch), then have a look at the following URL for Perl - Style regular expressions: http://www.cs.tut.fi/~jkorpela/perl/regexp.html.

Example:

- `WUInstall /install /nomatch "(Internet Explorer).*8 "`: Install all Updates except the ones for the Internet Explorer 8.

## Is it possible to deploy IE8 without user interaction using WuInstall?

With this command, you should get the absolute minimum of user interaction (=none) when installing IE 8:

```
wuinstall.exe /install /autoaccepteula /match "Internet Explorer 8"
/disable_ie_firstruncustomize /quiet /disableprompt
```

- /install ... installs the update
- /match "Internet Explorer 8" ... Filters only for Updates with title matching "Internet Explorer 8"
- /autoaccepteula ... Accepts any EULAs automatically (suppress user interaction on that)
- /disable_ie_firstruncustomize ... Disables the internet explorer 8 set up dialog.
- /quiet ... The update-installer is forced to do an installation without user interaction
- /disableprompt ... Disable showing source prompts to a user when installing the updates

## Why does WuInstall crash/generate failures during the download or installation of updates?

Are you using the show_progress option? This feature is not recommend for common usage. It might cause under certain circumstances on some machines problems.
It is only intended for manual testing on single machines. Please, try it without this option.

## WuInstall doesn't install updates properly when starting with "run as

### administrator", why?

When using Windows XP it might cause problems using the "run as" command together with WuInstall. To prevent this you should use the /quiet and /disableprompt option on the end of your command.

### Windows 7 Service Pack 1 installation fails, why?

Do you use the option /quiet when executing WuInstall? The installation of Windows 7 Service Pack 1 with the option /quiet could cause some error. Please try to execute WuInstall without /quiet.

### Installation of updates with WuInstall does not work on my Windows XP machine, why?

If you get unexplainable errors (like for example 0x80240022 - WU_E_ALL_UPDATES_FAILED) installing updates with WuInstall on Windows XP - even though running it with admin rights - try the following workaround:

If you are using the educational version, download the fully functional Trial Version of WuInstall Pro. Then just add the parameters /quiet /disableprompt to your WuInstall call - this should fix the issue.

Also take a look at this faq.

### Does WuInstall work with Windows 8 / Server 2012?

WuInstall 2.0 is optimized and certified for Windows 8 / Server 2012, but compatible with all Windows Versions. Therefore we highly recommend all users of WuInstall 1.x to upgrade to WuInstall 2.0.

## 2. Error codes

### WuInstall exits with error code 0x80072EFD, why?

This error usually indicates a connection problem. There is no connection to the WSUS or to the Windows Update Server. Probably, the WSUS - address is just configured incorrectly or there are proxy problems.

See also http://support.microsoft.com/kb/836941/en-us

### WuInstall exits with error code 0x8024400A, why?

This Problem sometimes occurs with new Windows XP SP 3 Machines, when the Windows Update Agent is not properly installed. Just download according to http://support.microsoft.com/kb/949104 the latest Version of the Windows Update Agent and install it.

### WuInstall exits with error code 0x80240044, why?

This error code occurs when administrator privileges are necessary to perform the wanted operation. Get sure that you run WuInstall with an administrator user.
For further information about error codes see please Error Codes.

## 3. WuInstall PRO Licensing

### How is the WuInstall license price calculated?

You can calculate the price of your WuInstall license using the license calculator. The price depends on the number of machines which you want to patch within your organization using WuInstall.

This is the current price model:

- max. 100 machines: USD 290 – WuInstall PRO Basic
- max. 250 machines: USD 490 – WuInstall PRO Small
- max. 500 machines: USD 790 – WuInstall PRO Medium
- max. 1000 machines: USD 1290 – WuInstall PRO Large
- max. 2000 machines: USD 2190 – WuInstall PRO Corporate
- max. 4000 machines: USD 3890– WuInstall PRO Enterprise
- more than 4000 machines: Please contact us at wuinstall@hs2n.at for an individual offer.

The prices above are yearly subscription fees. The license is always valid for one year from the date the subscription fee was paid.

For educational organizations (schools, universities, ...) the price of WuInstall PRO Basic applies, regardless of the number of servers or workstations.

## How do I get updates of WuInstall PRO?

You can get the latest WuInstall 1.x version by dropping an email to support@wuinstall.com. All WuInstall 2.0 customers have access to a customer portal, where they can download the latest version of WuInstall 2.0 at any time.

## How long is the WuInstall license valid?

The WuInstall license is valid for the licensed number of machines for one year from the date the subscription fee was paid.