

NAME

wimupdate – Update a WIM image

SYNOPSIS

wimupdate *WIMFILE* [*IMAGE*] [*OPTION...*] [*< CMDFILE*]

DESCRIPTION

wimupdate, or equivalently **wimlib-imagex update**, modifies the specified *IMAGE* in the Windows Imaging (WIM) archive *WIMFILE* by adding, deleting, or renaming files or directories in it.

IMAGE specifies the image in *WIMFILE* to update. It may be the 1-based index of an image or the name of an image. It may be omitted if *WIMFILE* contains only one image. You can use **wiminfo**(1) to list the images contained in *WIMFILE*.

The modifications to perform on the WIM image are specified as a sequence of commands, one per line, read in a text file from standard input. It is recommended that standard input be redirected from a file (*CMDFILE*), as shown above, rather than typing in commands interactively. Alternatively, to specify a command directly on the command line, see the **--command** option.

AVAILABLE COMMANDS

This section documents the commands that may appear in the *CMDFILE* described above.

add [*OPTION...*] *SOURCE DESTINATION*

Add a file or directory tree to the WIM image. *SOURCE* must specify the path to a file or directory on your filesystem. *DESTINATION* must specify the path inside the WIM image at which to add the file or directory tree.

If *DESTINATION* names an existing directory in the WIM image, then *SOURCE* must also name a directory. This causes the contents of the *SOURCE* directory to be added to the *DESTINATION* directory.

If *DESTINATION* names an existing nondirectory file in the WIM image, then *SOURCE* must also name a nondirectory file. By default, this causes the *DESTINATION* file to be replaced with the *SOURCE* file. Or, with **--no-replace** specified, this generates an error.

If *DESTINATION* does not exist in the WIM image, then any prerequisite directories are created as needed to add the *SOURCE* at that location.

The **add** command supports a subset of the options accepted by **wimcapture**; namely, **--dereference**, **--unix-data**, **--no-acls**, and **--strict-acls**. See **wimcapture**(1) for explanations of these options.

In addition, the **add** command supports the **--no-replace** option, which causes the **add** command to refuse to overwrite existing nondirectory files in the WIM image.

delete [*OPTION...*] *PATH*

Delete a file or directory tree from the WIM image. *PATH* must specify the path inside the WIM image of the file or directory tree to delete.

The available options for the **delete** command are:

--force

Do not issue an error if the path to delete does not exist.

--recursive

Delete the file or directory tree recursively; if not specified, an error is issued if the path to delete is a directory.

rename *OLD_PATH NEW_PATH*

Rename a file or directory tree inside the WIM image. *OLD_PATH* must specify the old path of the file or directory tree inside the WIM image, and *NEW_PATH* must specify the new path for the file or directory tree. This command follows the semantics of the POSIX **rename** (3) function; in particular, a pre-existing file at *NEW_PATH* will be deleted if present, except in certain cases such as attempting to rename a directory to a non-directory, which is not allowed.

There are no options available for the **rename** command.

OPTIONS

The following options are accepted on the command line by **wimupdate** itself:

--dereference

Use **--dereference** for all **add** commands.

--unix-data

Use **--unix-data** for all **add** commands.

--no-acls

Use **--no-acls** for all **add** commands.

--strict-acls

Use **--strict-acls** for all **add** commands.

--no-replace

Use **--no-replace** for all **add** commands.

--config=FILE

Set the capture configuration file for all **add** commands. See the description of this option to **wim-capture**(1).

--force

Use **--force** for all **delete** commands.

--recursive

Use **--recursive** for all **delete** commands.

--check

Before updating the WIM, verify its integrity if it contains extra integrity information. Also include extra integrity information in the updated WIM even if it was not present before.

--threads=NUM_THREADS

Number of threads to use for compressing newly added files. Default: autodetect (number of processors).

--rebuild

Rebuild the entire WIM rather than appending the updated data to the end of it. Rebuilding the WIM is slower, but will save a little bit of space that would otherwise be left as a hole in the WIM file.

See **wimoptimize**(1) for a more customizable way of rebuilding (and optionally recompressing) a WIM file. If running **wimoptimize** after **wimupdate**, there is no need to specify **--rebuild** to **wimupdate**.

--command=STRING

Instead of reading update commands from standard input, read a single update command directly from the string *STRING* specified on the command line. This option cannot be provided more than one time and cannot be used to specify more than one update command. Note that the *STRING*, as well as any paths containing spaces within the *STRING* must be appropriately quoted. If running from cmd.exe on Windows, you should use double quotes for the outer quotes and single quotes for the inner quotes. Example:

```
wimupdate boot.wim 1 --command="add 'C:\My Dir' '\My Dir'"
```

This option is provided for convenience only. Do not execute **wimupdate** multiple consecutive times, each time passing the **--command** option! This is inefficient. Instead, generate an update command file and provide it (on standard input) to a single invocation of **wimupdate**, as explained in this document.

--wimboot-config=FILE

If this option is specified, no commands shall be read from standard input, and instead the following command shall be executed:

add *FILE* /Windows/System32/WimBootCompress.ini

This sets *FILE* as the WIMBoot configuration file for the image. The [PrepopulateList] section of this file specifies path globs that shall not be extracted as WIMBoot pointer files (perhaps due to being needed early in the boot process). See the documentation for the **--wimboot** option of **wimapply**(1) for more information.

--unsafe-compact

Compact the WIM archive in-place and append any new data, eliminating "holes". This is efficient, but in general this option should *not* be used because a failed or interrupted compaction will corrupt the WIM archive. For more information, see the documentation for this option in **wimoptimize**(1).

NOTES

wimupdate can be viewed as redundant with **wimmountrw**, since a WIM image can also be updated by mounting it read-write. However, **wimupdate** works on all platforms including Windows, whereas **wimmountrw** only works on Linux.

Symbolic links inside a WIM image are not dereferenced when being interpreted. So, for example, if you have a WIM image that contains a symbolic link `"/Documents and Settings" -> "/Users"` where `"/Users"` is a directory, then a subdirectory named `"Public"` in this directory must be specified as `"/Users/Public"` rather than `"/Documents and Settings/Public"`.

All paths to files or directories within the WIM image must be specified relative to the root of the image. However, the leading slash is optional, and both forward slashes and backslashes are accepted. In addition, on Windows, the paths are by default treated case-insensitively, while on UNIX-like systems, the paths are by default treated case-sensitively. The default case sensitivity may be changed by setting the **WIM-LIB_IMAGEX_IGNORE_CASE** environmental variable to 0 or 1.

The command file (*CMDFILE*) is parsed by **wimupdate** itself and not by the system shell. Therefore, its syntax is limited. However, comment lines beginning with `'#'` are allowed, and it is also possible to quote arguments with whitespace inside them.

On UNIX-like systems, you cannot use **wimupdate** to add files to an image directly from an NTFS volume using `libntfs-3g`, even though **wimcapture** supports capturing a full image this way.

Except when using **--unsafe-compact**, it is safe to abort a **wimupdate** command partway through; however, after doing this, it is recommended to run **wimoptimize** to remove any data that was appended to the physical WIM file but not yet incorporated into the structure of the WIM, unless **--rebuild** was specified, in which case you should delete the temporary file left over.

EXAMPLES

All the examples below show the update command file to be created as well as the **wimupdate** command to run to perform the updates.

Delete two files from a WIM image:

```
update_commands.txt:
    delete /setup.exe
    delete /sources/setup.exe

$ wimupdate boot.wim 2 < update_commands.txt
```

Add some files and directories to a WIM image. Note that the first path of each **add** command specifies the files to add, while the second path of each **add** command specify the locations at which to add them inside the WIM image:

```
update_commands.txt:
    add somedir    /dir
    add somefile   /dir/file

$ wimupdate boot.wim 2 < update_commands.txt
```

Rename a file inside a WIM image.

update_commands.txt:

```
rename /dir_in_wim/oldfile.txt /dir_in_wim/newfile.txt
```

```
$ wimupdate boot.wim 2 < update_commands.txt
```

Using additional features, such as comments, options, and overlays, and including extra integrity information in the updated WIM:

update_commands.txt:

```
#  
# This file specifies some changes to make to a WIM image.  
#
```

```
# Add a new directory containing files I want in the image.  
# The quotes are necessary because the directory name  
# contains a space.  
add "My Directory" "/My Directory"
```

```
# Add the contents of "Another Directory" to the  
# "/My Directory" we just created in the WIM image. Since  
# the destination path already exists, this performs an  
# overlay.  
add "Another Directory" "/My Directory"
```

```
# Rename some file for some reason.  
rename /dir_in_wim/oldfile.txt /dir_in_wim/newfile.txt
```

```
# Delete an unwanted directory.  
delete --recursive /Users/Me/Documents/Junk
```

```
$ wimupdate boot.wim 2 --check < update_commands.txt
```

SEE ALSO

wimlib-imagex(1) **wimcapture(1)** **wiminfo(1)** **wimmountrw(1)** **wimoptimize(1)**