
HTTPUP - USER MANUAL

One way sync over http

Author

Johannes Winkelmann, jw@tks6.net

Table of Contents

1. Introduction	1
2. How to update your local copy	1
2.1. Examples	1
3. How to create a repository	1
3.1. Ignoring files	1
3.2. Recommended setup	2
3.3. Examples	2
4. example supfile for the httpup ports driver	2
5. Copyright and licensing	2
5.1. Copyright	2
5.2. License	2

1. Introduction

httpup is a simple program to synchronize data in one direction. It uses HTTP as transport, using `libcurl` as backend.

If you just like to synchronize a httpup-collection, it should be enough to read the next section only.

2. How to update your local copy

Obviously you need to have a working installation of httpup.

To synchronize a httpup repository with a local directory, just pass those two arguments to the **httpup** command:

```
$ httpup URL target-directory
```

2.1. Examples

```
$ httpup http://www.hta-bi.bfh.ch/~winkj/files/crux/ports-httpsync/ /tmp/ports
```

3. How to create a repository

To create repository, either run **httpup-repgen** in the directory where your repository is, or pass an additional parameter where **httpup-repgen** should create a repository.

```
$ httpup-repgen [directory]
```

3.1. Ignoring files

To avoid inclusion of certain files in the repository, httpup-repgen has a mechanism to ignore certain file patterns. It checks two files for ignore expressions: first `.httpup-repgen-ignore` in the repository base directory, then `$HOME/.httpup-repgen-ignore`. These files can contain regular expressions which will be fed to **grep(1)** to filter out matches. If for example you'd like to ignore all files ending on

'tar.gz', you'd add the following to your `.httpup-ignore-expression`:

```
\.tar\.gz$
```

Every line can contain one expression, so in the end, a typical ignore file might look like the following:

```
\.svn$
\.tar\.gz$
\.tgz$
\.tar\.bz2$
\.zip$
^CVS$
^.*CVS$
/CVS/. *
^CVS/. *
.*~$
```

The ignore file is ignored automatically.

3.2. Recommended setup

While this is no requirement, the following setup has proven to help avoiding errors during ports publication: First of all, define a location where your ports live in your local file system (I keep mine in `$HOME/build/upload`). Define some ignore expressions (see the previous paragraph for more information), to make sure you're only including the files you really want. You may keep this directory under revision control, e.g. using CVS, subversion, monotone etc.

Second, choose a tool to synchronize this repository to the remote server. There are a few tools available with support for different network protocols: `rsync`, `unison`, `weex`. There's also a dedicated tool called `htup` [<http://members.shaw.ca/nick.steeves/ports/htup>] written by Nick Steves which automates those steps.

3.3. Examples

Create a repository in the current directory:

```
$ httpup-repgen
```

Create a repository in `/tmp/repo`:

```
$ httpup-repgen /tmp/repo
```

4. example supfile for the httpup ports driver

```
#
# /etc/ports/jw.httpup: Johannes Winkelmann's port collection
#
ROOT_DIR=/usr/ports/jw
URL=http://www.hta-bi.bfh.ch/~winkj/files/crux/ports-httpsync/
# End of file
```

5. Copyright and licensing

5.1. Copyright

Copyright (c) 2002-2003 by Johannes Winkelmann, jw@tk6.net

5.2. License

httpup is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.