



# File Synchroniser

# Contents

**Contents** ..... 2

1 **Description** ..... 3

2 **Server (or Master) file sharing** ..... 3

3 **Command Line Parameters** ..... 3

4 **Pre Sync App** ..... 5

5 **Post Sync App** ..... 5

6 **File exclusions** ..... 5

7 **Logs** ..... 6

8 **CC\_DEVELOPER Checking** ..... 6

9 **Notes** ..... 6

10 **Bug list** ..... 6

11 **Associated Applications** ..... 6

# 1 Description

This is a utility for synchronising file and directory structures.

**Important note:**

This is an aggressive file copying engine it will completely synchronise a slave directory with the master regardless of file dates, and will delete files on the slave that do not exist on the master. Be very careful in setting this program up!

**Important note:**

This application is meant for a client/driver machine on a BNCS system rather than as a generic file copying mechanism.

# 2 Server (or Master) file sharing

File copying is done using a Windows networking and requires shares (with appropriate permissions) to be set up. This application must have read access to the master machine. A completely open share is undesirable; the best solution is for the client machine to have an account on the file server so that credentials do not have to be supplied to access the data.

The usual cause of failure of a file sync operation is permissions to the specified shares.

# 3 Command Line Parameters

```
filesync /source=<directory> /dest=<directory> [/reservesource=<directory> /delete  
/postsync=<application> /presync=<application> /quiet /static /exclude]
```

Parameter	Optional	Notes
/source=<sourcedir>		the master copy of the data None of the files in this path are changed.
/reservesource=<sourcedir>		Path to the reserve server
/dest=<destdir>		the destination / slave copy After the files are synchronised this path will be an accurate copy of the master_directory.
/delete	Optional	if set to "true" files are deleted from the slave_directory if they do not exist on the server.  if "false" (the default) files are updated from the master but redundant files never deleted. This is marginally safer if you manage to get the directories muddled up

/postsync=<application>	Optional	<p>The application to run once the file sync process is complete.</p> <p><b>Note:</b> filesync does not close down until this application has terminated (unless it failed to run it or wasn't present on the command line).</p> <p>You don't need to specify an absolute path if this application resides in the path of the machine.</p> <p>This could be v2launch.exe but ensure that the QuitWhenDone flag is set.</p>
/presync=<application>	Optional	
/quiet	Optional	This disables the CC_DEVELOPER flag checking
/exclude=<dirlist>	Optional	<p>Comma delimited list of top level directories to exclude. Typically for V4.5 installations this is:</p> <p>/exclude=source,temp,docs,data,install,backup,logs</p> <p>so when:</p> <p style="padding-left: 40px;">/source=r:\bncs</p> <p>the directories</p> <p style="padding-left: 40px;">r:\bncs\source</p> <p style="padding-left: 40px;">r:\bncs\temp</p> <p style="padding-left: 40px;">etc.</p> <p>would be excluded from the sync along with all others specified in this list</p>
/static	Optional	<p>Forces the use of the filelist.sync instead of dynamically determining the list of files dynamically from the server each time</p> <p>This is only any good if this list of files is maintained and whilst it is maintained by the V4.5 configuration server manual changes to the file system are not captured.</p>

Example: the following command line runs a batch file "rbatch.bat" before starting file synchronisation, then synchronises files on the local machine (c:\my path\v4.5) with a server (\\bncs\_001\c\$\bncs\v4.5) deleting slave files appropriately and running a program called "obatch.bat" when the update is complete.

```
filesync /source=\\bncs_001\c$\bncs\v4.5 /dest="c:\my path\v4.5" /delete
/postsync=obatch.bat /presync=rbatch.bat
```

**Note:** the optional use of quotes to ensure that paths with spaces in aren't misinterpreted as additional command line parameters.

## 4 Pre Sync App

The idea of the pre-sync batch file is to perhaps do default actions whether or not the sync operation is successful (such as removing temporary files to force a resync with the server).

## 5 Post Sync App

The idea of the post-sync application is to run a batch file or whatever to do any post-synchronise actions. An example of this might be to move selected files that have just been copied within the BNCS file structure to the Windows directory for those applications that need it, as the Windows directory itself cannot be synchronised (or at least it could but your machine would never work again!).

e.g. based on the above example this batch file takes the contents of the "windir" subdirectory and copies it to Windows

contents of file postsync.bat:

```
xcopy %BNCS_ROOT%\windir\*.* %WINDIR% /c/i/f/d/y
```

(in this example the environment variables for BNCS\_ROOT and WINDIR are used rather than explicitly specified paths)

## 6 File exclusions

The command line accepts a parameter of directories not to sync.

There is a second means of excluding files by pattern from an external configuration.

Configuration is to be found here:

```
%CC_ROOT%\%CC_SYSTEM%\config\bncs_filesync.xml
```

and is of the following format:

```
<?xml version="1.0" ?>
<exclusions>
  <regex wildcard="true" >*.tlog</regex>
  <regex case="true" wildcard="true" >*.sdf</regex>
</exclusions>
```

Each entry is a regular expression (as the Qt toolkit it's based on knows them).

The `case` and `wildcard` attributes are entirely optional but if set to "true" enable case sensitive pattern matching and wildcard matching respectively.

If you wanted to exclude all files whose extensions where ".cpp" either of these entries would work:

```
<regexp wildcard="true" >*.cpp</regexp>
```

```
<regexp>.cpp$</regexp>
```

## 7 Logs

This applications writes a daily log file of activity in standard log file format which can be found in:

```
%CC_ROOT%\%CC_SYSTEM%\logs\FileSync\yyyymmdd.log
```

for example

```
c:\bncs\v4.5\logs\FileSync\20040816.log
```

It's anticipated that an external application is used to delete unwanted log files.

## 8 CC\_DEVELOPER Checking

If the CC\_DEVELOPER environment variable is set to "true" this application throws up a message box to inform the user that they are about to synchronise part of their file system. This is just a sanity check so that you can't wipe out a file structure that you didn't mean to. This checking can be disabled by specifying "/quiet" on the command line.

## 9 Notes

This application can't overwrite files that are currently in use such as binaries (DLLs or EXEs) or files locked by applications.

Synch'ing an entire tree assumes that there is no local settings or temporary data in the slave tree.

## 10 Bug list

Slave file dates are synchronised with the master. However there are likely to be problems with running this application over a daylight saving change (March/October). This might involve a complete resync of the slave – not disastrous but unnecessarily time consuming.

## 11 Associated Applications

For efficiency the master machine may use a pre-determined list of files rather than have the file list generated every time. The file is called filelist.sync and is located in the root of the server (usually "%CC\_SERVER\_ROOT%\%CC\_SYSTEM\filelist.sync").

There is a simple utility to generate this list called *bncs\_filesync\_list*.

Atos  
4 Triton Square  
Regent's Place  
London NW1 3HG, UK