EasySync

Author
Keith Merrington

Purpose

- The purpose of EasySync is to syncronise two directories with each other.
- Optionally support file backup
- Normally the two directories are on different machines.
- Simple Interface, Graphical

How to keep it easy - 1

Design Decision:

- Graphical ➔PM
  - Front end OR new program ?
    - Make a graphical front end for an existing program such as DSYNC/RSYNC, and accept limitations for interface

- Make a new program!

How to keep it easy - 2

Design Decision:

- Simple ➔Single program
- Sync ➔ bi-directional transfer
- Backup ➔ uni-directional transfer
- Use Drive letters to select attached machine.
- Allow an automatic synchronisation method
- Multilingual ➔ to be implemented
- Use ICONS to limit problems with button text

How to keep it easy- 3

Design Decision:

- Provide bubble text and help
- Provide user feedback on what's happening
- Allow full automatic synchronisation
- Allow manual synchronisation

  - determine what situations and actions are needed

What possible situations

- Files same Size, Date and Time but different
- Directory Only Exists at Source
- Source has changed, destination was deleted
- Destination File was Deleted after last sync
- Dir Only Exists at Destination
- File Exist Only at Source
- File Exist Only at Destination *
- Files same Size, different date and or Time
- Files Identical, different time stamps
- Both Files have changed since last sync
- Files have same Date & Time, different Size
- Source File is Newer
- Source File is Older *
- Errors

How to determine if a file is different

Design Decision:

Files are not necessarily identical when they both have the same size, and modification dates.

- Use MD5 checksums
  - Efficiency - Only do this when a file changes
    - How - save both checksum and file, date and size to which it refers.
    - Where - Use an EA to store checksum, file, date and size and the info (file is in principle remains the same)
    - What – name the EA ‘MD5CHECKSUM’
- Make this optional since this can be time consuming.

What Actions Are Possible

The basic file actions are:

- Create / Delete
- Overwrite / Copy / Move
- Set date stamp

- The basic directory actions are:
  - Create / delete / move
- User
  - Ask /Assume

Requirements

- In order to synchronise the following information is necessary:
  - Method
  - Directory names to be synchronised
  - Actions to be taken
  - Automatic or Manual

Method

- There two synchronisation methods available which may be selected as required.
  - Uni-directional (Master/Slave)
  - Bi-Directional (Equals)

How to specify the directories

Design Decision:

- Allow file name input in multiple ways:
  - Keyboard (+ file name completion)
  - File browser
  - From a file ➔ Script file.
- Remember recently used ➔Drop down list box.

Directory Names

- The fully qualified path name is required for both the *‘Local’* and the *‘Local/Remote´* directories.
- They may be specified :
  - Directly using the keyboard
  - Indirectly by using the browser
  - From a script file

What to do when a file is deleted

Design Decision:

- Detect which files have been deleted
  - Make list of files present in both directories after successful synchronisation
  - Keep this list in the SyncDel directory belonging to that file (this avoids drive letter confusion!)
  - Check prior to synchronisations if a list exists and what files are now present, files missing from list have therefore been deleted
  - Keep list in a special directory
  - Store list on the machine belonging to that path

Actions

- During synchronisation the action to be taken must be specified.
- The various actions that can be taken are
  - Prompt
  - Skip
  - Transfer
  - Delete
  - Create directory
  - …

Actions /Manual _ Automatic

- Allow manual and automatic syncronisation
- Use two action tables
  - Manual action list
  - Automatic action list

Automatic action list

- The automatic action list is for unattended operation. As such it is not possible to select prompt. If an action cannot be determined before hand,
  - it may be skipped
  - it may be skipped and the anomaly counter incremented

Synchronising

- Before the synchronising action is initiated both windows will have been filled with the contents of each directory and subdirectories.
- An initial transfer list is displayed in the middle column indicating what files are different and what would normally be transferred.

Synchronising

Design Decision:

- Provide an indication of file differences before starting synchronisation.
- Make a log of all transactions
  - Use date and time to create checksum (7 digits + 1) to make unique name that is 8.3 complient

Synchronising

- Several radio buttons are available to show
  - All files
  - files that have been deleted.
  - newer files
  - older files
  - missing files
  - Ignored files

Starting the synchronisation

- Pressing the Sync button will start the synchronisation procedure and for each file situation the appropriate action is taken.
- If prompt was specified then a dialog is displayed with the reason for the prompt plus both files details and several action buttons so that the operator may determine what action to take.
-

What to do when a file is deleted/overwritten

Design decision

- Make a copy of the file being overwritten or deleted.
  - Files are moved to a directory (SyncDel).
  - Allow multiple versions of the same file
- Keep overhead to a minimum:
  - Files are only moved from the current drive to the directory on that drive.
  - This directory is created on each logical drive as required.
- How to handle filenames having a maximum length
  - Use a hash to generate a unique name of limited length
  - Save the original name in a text file

The Vault

When ever a syncronisation action would result in a file being

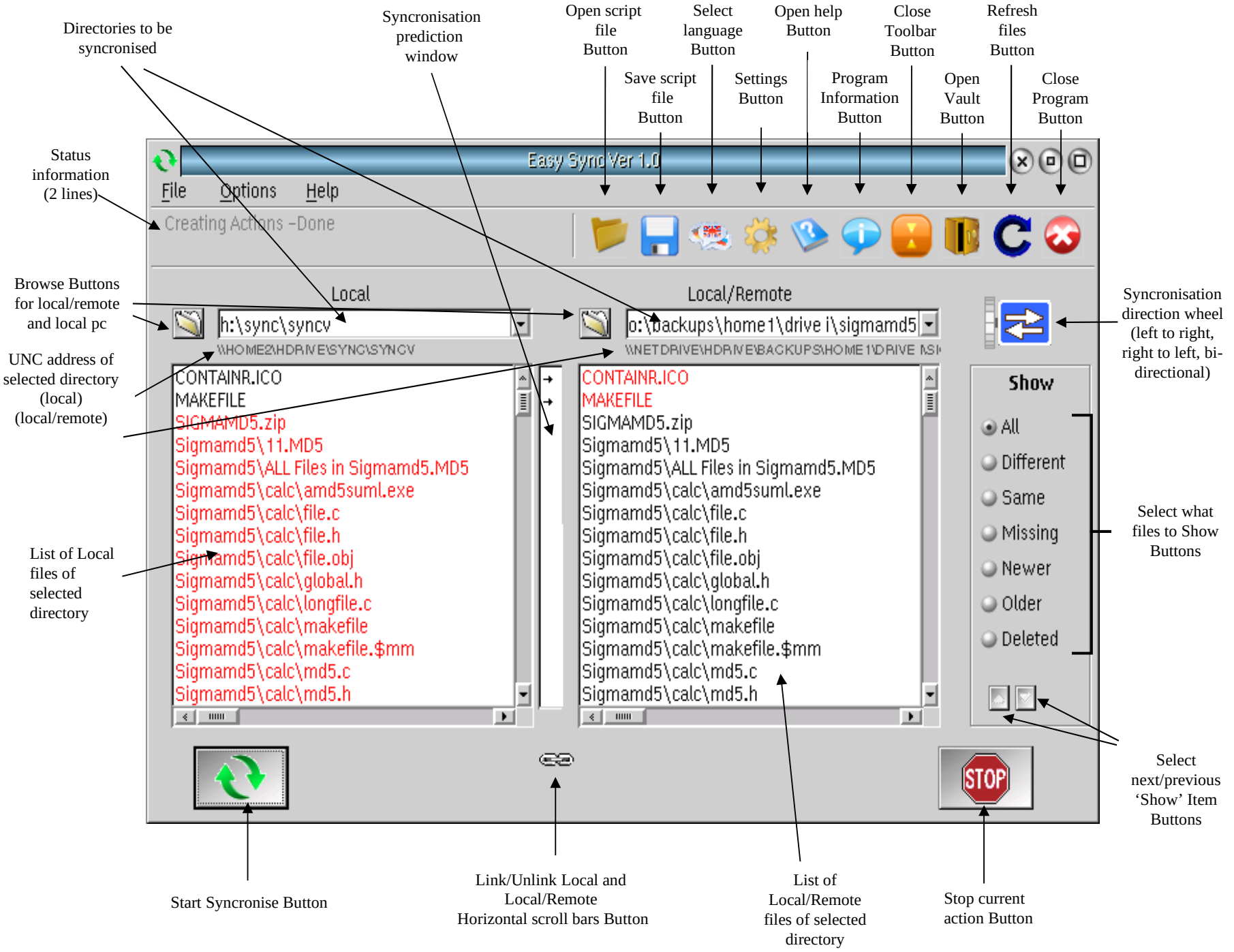
- Overwritten
- Deleted

That file is moved to the vault.

- The vault contains all files previously deleted-or overwritten.

EasySync

Demo


Directories to be syncronised
Syncronisation prediction window
Open script file Button
Select language Button
Open help Button
Close Toolbar Button
Refresh files Button
Save script file Button
Settings Button
Program Information Button
Open Vault Button
Close Program Button
Status information (2 lines)
Easy Sync Ver 1.0
File
Options
Help
Creating Actions -Done
Browse Buttons for local/remote and local pc
Local
Local/Remote
h:\sync\syncv
o:\backups\home1\drive i\sigmamd5
\\HOME2\HDRIVE\SYNC\SYNCV
\\NETDRIVE\HDRIVE\BACKUPS\HOME1\DRIVE I\SI
UNC address of selected directory (local) (local/remote)
Syncronisation direction wheel (left to right, right to left, bi-directional)
CONTAINR.ICO
MAKEFILE
SIGMAMD5.zip
Sigmamd5\11.MD5
Sigmamd5\ALL Files in Sigmamd5.MD5
Sigmamd5\calc\amd5suml.exe
Sigmamd5\calc\file.c
Sigmamd5\calc\file.h
Sigmamd5\calc\file.obj
Sigmamd5\calc\global.h
Sigmamd5\calc\longfile.c
Sigmamd5\calc\makefile
Sigmamd5\calc\makefile.$mm
Sigmamd5\calc\md5.c
Sigmamd5\calc\md5.h
List of Local files of selected directory
Show
All
Different
Same
Missing
Newer
Older
Deleted
Select what files to Show Buttons
Select next/previous 'Show' Item Buttons
STOP
Start Syncronise Button
Link/Unlink Local and Local/Remote Horizontal scroll bars Button
List of Local/Remote files of selected directory
Stop current action Button

Suggestions

What's next

1. Change input windows to support unlimited number of files
2. Improve Vault with drag and drop
3. Add "Ignore Paths"
4. Add file transfer verification
5. Add file querying before syncronisation
6. ?