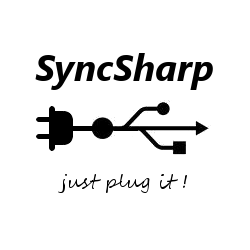
**Project name: SyncSharp**

****

**Team 13: EXCALIBUR**

**Azhar**

**Chris**

**Hong Lei**

**Jia Yuan**

**Tian Shuang**

**Yew Kang**

***Table of content***

***Chapter 0: Introduction***

**Typographic Conventions 1**

**Introduction 2**

**Product Features**

**System Requirement**

***Chapter 1: Developers Guide***

**Part 1: Design Methodology**

Top-Down Incremental Compilation Flow

**Part 2: Design Flow**

System Architecture

Domain Model Analysis

Class Diagram

Sequence Diagram

Use cases

Activity Diagram

***Chapter 2: Users Guide***

**Part 1: Product Description**

**Part 2: Using SyncSharp**

On first run 13

Creating/deleting new Sync task 12

Modifying new Sync task 12

Review sync result 12

How to import/export Task 12

View Logs 25

PlugSync 31

**Part 4: Tips on Using SyncSharp**

Inserting Paper 33

**Part 5: Technical Information**

Specifications 53

Radio & Television Interference 55

**Part 6: Index**

***Chapter 3: Glossary***

***Chapter 4: Appendix***

**Typographic Convention**

All typographic conventions used in this documentation are shown below:

|  |  |  |
| --- | --- | --- |
| Visual cue | **Meaning** | |
|  | |  |
|  |  | |
|  |  | |
|  |  | |

**Chapter 0**

**Introduction**

**Introduction**

This documentation is written in response to develop a file (and folder) synchronization tool for CS3215 Software engineering Project.

File/Folder synchronization tools are commonly uses who want to synchronize files and folders across multiple computers. They allow users to modify and update files and folders in two or more locations through certain rules. Most of synchronization tools provide users a one-way sync, where files and folders are copied from the source location to the destination location, or two-way sync, where files and folders are copied in both locations so that files and folders are identical in both source and destination directories.

However, most of the sync tools that are available in the market required installation which may considered as a hassle to some users. It is because not all computers pre-installed with file synchronization software, and users may not be granted with administrative rights to install software, and this poses problems for users who need to perform file synchronization.

In order to solve the abovementioned problems, our team has developed a file (and folder) synchronization tool called SyncSharp which provides users with a streamline file (and folder) synchronization operation and installation free application.

**Product Features**

SyncSharp is a Windows based application which allows users to sync files and folders between multiple computers through an immediate device with no installation required. Our product provides a one-way synchronization action since many users may just want to use file synchronization tool as backup purpose. It can be very helpful when users do not know which files already exist in the backup storage. The tool will automatically check for any new updates and perform synchronization action to files and folders while users perform other tasks on the computer. Our product also provides two-way synchronization action which can automatically detect which files and folders have been modified and perform synchronization actions between source and destination directory based on pre-determined user preferences. In addition, SyncSharp is easy to use and supports all the basic features of a typical synchronization tool. SyncSharp offers users with automated synchronization with minimal user interaction.

* Create, edit and delete synchronization profiles
* Import/export synchronization profiles
* Ability to use environment variables in folder paths
* User settings
* Compare source and target directories
* Preview file contents before synchronization action
* Perform 2-way synchronization between the file/folder pairs
* Set inclusion/exclusion filters
* Backup files
* Generate log file after each synchronization operation

**System Requirements**

|  |  |
| --- | --- |
| Operating System: | Windows 2000,  Windows 2003,  Windows XP,  Windows Vista,  Windows 7[[1]](#footnote-1) |
| PC Configuration requirement | 128 RAM or more  X MB of hard disk space. |

**Support and feedback**

**Technical support**

For technical support, please contact us by email at [CS3215-13@gmail.com](mailto:CS3215-13@gmail.com)

**Feedback**

If you have any comments or suggestions for the next release, please direct them to [cs3215-13@gmail.com](mailto:cs3215-13@gmail.com). Your feedback is highly important for us. In order to get idea of how to make SyncSharp a better product for you, the current release is highly influenced by comments from users.

**Chapter 1**

**Developer Guide**

**Developers Guide**

System Architecture



Components Description

|  |  |
| --- | --- |
| Component | Description |
| GUI | Provides the interface between users and application. |
| ATD | Provides automated testing of the application functionalities during development. |
| SyncSharp Logic | Receives input from GUI component and initiates a response by making function calls to various sub-components. |
| Detector | Evaluates the changes on the designated folders / files based on the last synchronization operation which stores a small amount of information (called metadata). Metadata captures a snapshot of every file and folders’ state. Detector then passes a list of files to the reconciler to perform synchronization. |
| Reconciler | Performs file synchronization on the list of files obtained from the Detector. The file synchronization operation is based on the analyzed results. In the rise of conflicting updates, pre-determined users’ preferences will be used to resolve the updating conflicts. The summary of the updates will be passed to the Logger. Reconciler then updates the metadata of the replica. |
| Filter | Provides a list of filter rules that will be used by Detector for files retrieval. |
| Sync Profile | Stores the machine identity and contains a list of SyncTask associated with the profile. |
| SyncTask | Defines the pair of folders to be used for synchronization |
| TaskSettings | Stores all the configuration settings for each SyncTask. |
| Logger | Generates the summary of the file synchronization tasks. |
| FileUnit | Abstract representation of a file or folder. |

Domain Model

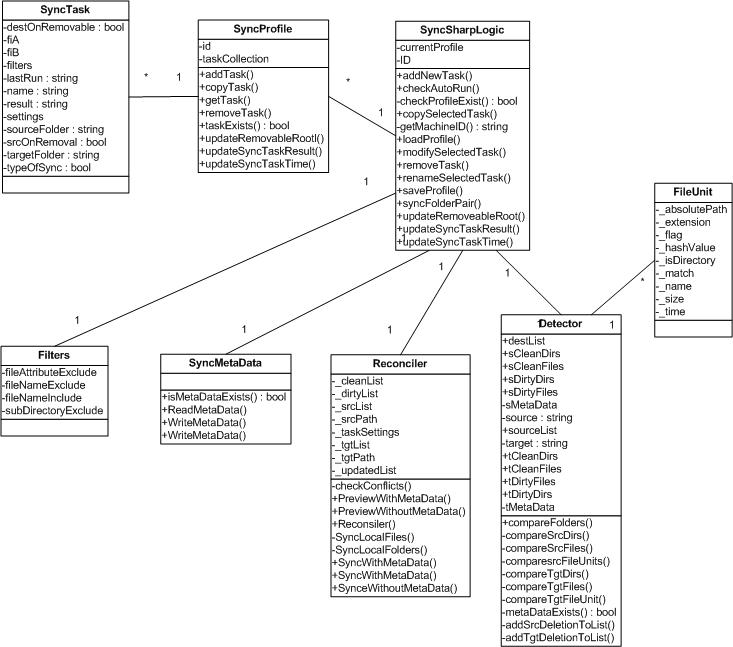


Overview

This team used top-down incremental design methodology in this project. We think that this is the best methodology for it. It is because incremental design methodology allows us to preserve what we have done and save compilation time by taking previous compilation results so that only the portion that has just been modified are compiled each time.

In the incremental compilation flow, we wrote the codes and compiled them incrementally. It is much easier for us to fix bugs when we face them; and it can help to save our time by allowing us to modify the critical portions while processing other portions. Reconciler performs actual file synchronization on the source and destination directories obtained from the Detector. The file synchronization operation is based on the analyzed results. In the rise of conflicting updates, pre-determined users’ settings will be used to resolve the updating conflicts. The summary of the updates will be passed to the Logger. Reconciler then updates the metadata of the replica.

Class Diagram



Sequence Diagram



Use Case Diagram

The following table is a summary of all use cases.

|  |
| --- |
| No User Case |
| 1 Create Synchronization Tasks |
| 2 Edit Synchronization Tasks |
| 3 Delete Synchronization Tasks |
| 4 Run PlugSync |
| 5 Compare Source and Target Directories |
| 6 Perform 2-way Synchronization Between Source and Target  Directories |
| 7 Backup Files |
| 8 Restore Files |
| 9 View Source/Target Folders |
| 10 Export Synchronization Profiles |
| 11 Import Synchronization Profiles |
| 12 View Log Files |
| 13 View Help Files |

User Case(s) Description

|  |
| --- |
| Use Case Number: 1  Use Case Name: Create Synchronization Tasks |
| Pre-Conditions: SyncSharp is running and at the main window |
| Post-Conditions: System creates new Synchronization task and is displayed on the  main window |
| Actors: User, System |
| Main Success Scenario:   1. User clicks on “New” 2. System displays new SyncTask setup wizard 3. User enters name for SyncTask 4. System requests for SyncTask type: ‘Synchronize’ or ‘Backup’ 5. User selects SyncTask type 6. System requests path for Source and Target folder 7. User enters path for Source and Target folder 8. System creates new SyncTask and updates the main window   Extensions(s):  3a. User enters a name that already exists for a SyncTask  3a1. System displays name already exist error  Use case resumes from step 2.  5a. User did not select a SyncTask type before attempting to proceed  5a1. System prompts user to select a SyncTask type  Use case resumes from step 4.  7a. User enters non-existing/empty path for Source/Target folders  7a1. System prompts user to enter a valid path name  Use case resumes from step 6.  7b. User selects same path for Source/Target folders  7b1. System displays error that Source/Target folders cannot be the same  Use case resumes from step 6. |

|  |
| --- |
| User Case Number: 2  User Case Name: Edit Synchronization Tasks |
| Pre-Conditions: At least 1 SyncTask has already been created |
| Post-Conditions: SyncTask settings are updated and main window is updated to reflect  any changes |
| Actors: User, System |
| Main Success Scenario:   1. User selects a SyncTask from the main window and clicks on “Modify” 2. System displays the task setup window 3. User modifies the SyncTask settings as desired 4. System updates the SyncTask settings and updates the main window to reflect any changes   Extensions(s):  3a. User provides some invalid settings  3a1. System prompts user to correct any errors  Use case resumes from step 2. |

|  |
| --- |
| User Case Number: 3  User Case Name: Delete Synchronization Tasks |
| Pre-Conditions: At least 1 SyncTask has already been created |
| Post-Conditions: Select SyncTask is deleted and removed from the main window |
| Actors: User, System |
| Main Success Scenario:   1. User selects a SyncTask from the main window and clicks on “Delete” 2. System confirms with user to delete selected SyncTask 3. User selects ‘OK’ 4. System deletes selected SyncTask and removes it from the main window   Extension(s):  3a. User selects ‘Cancel’  Use case ends. |

|  |
| --- |
| User Case Number: 4  User Case Name: Run PlugSync |
| Pre-Conditions: At least 1 SyncTask has been created and PlugSync is enabled for this  SyncTask, SyncSharp is run from removable USB device, Computer’s  AutoPlay is enabled |
| Post-Conditions: Source/Target folder contents are synchronized |
| Actors: User, System |
| Main Success Scenario:   1. User inserts removable USB device 2. System automatically initiates 3. System retrieves a list of SyncTasks from current profile that has PlugSync enabled 4. System displays countdown that PlugSync is about to start 5. User waits for countdown period to end 6. System performs synchronization 7. System returns back to main window. Normal usage continues   Extension(s)  5a. User cancels PlugSync by clicking on “Back to Main”  Use case resumes from step 7 |

|  |
| --- |
| Use Case Number: 5  Use Case Name: Compare Source and Target Directories |
| Pre-Conditions: At least 1 SyncTask has been created |
| Post-Conditions: A window is displayed to the user that shows all the differences and  SyncActions that would be performed by synchronization |
| Actors: User, System |
| Main Success Scenario:   1. User selects SyncTask and clicks on ‘Analyze’ 2. System compares Source/Target folders and displays results to user   Extension(s):  2a. System determines that Source/Target folders are already synchronized, and displays message to user  User case ends |

|  |
| --- |
| Use Case Number: 6  Use Case Name: Perform 2-way Synchronization Between Source and Target Directories |
| Pre-Conditions: At least 1 SyncTask has been created, with ‘Synchronization’ type |
| Post-Conditions: Source/Target folder contents are synchronized |
| Actors: User, System |
| Main Success Scenario:   1. User selects SyncTask and clicks on ‘Synchronize’ 2. System proceeds to synchronize the Source/Target folders 3. System updates “Successful”, and last run time in the main window for the selected SyncTask   Extension(s):  2a. System encounters error during synchronization  2a1. System updates “Unsuccessful”, and last run time in the main window for  the selected SyncTask  Use case ends |

|  |
| --- |
| Use Case Number: 7  Use Case Name: Backup Files |
| Pre-Conditions: At least 1 SyncTask has been created with ‘Backup’ type |
| Post-Conditions: Any changes made to files/folders on Source directory will be updated  on Target directory |
| Actors: User, System |
| Main Success Scenario:   1. User selects SyncTask and clicks on ‘Backup’ 2. System proceeds to backup the Source folder to the Target folder 3. System updates “Successful”, and last run time in the main window for the selected SyncTask   Extension(s):  2a. System encounters error during backup  2a1. System updates “Unsuccessful”, and last run time in the main window for  the selected SyncTask  Use case ends |

|  |
| --- |
| Use Case Number: 8  Use Case Name: Restore Files |
| Pre-Conditions: At least 1 SyncTask has been created with ‘Backup’ type |
| Post-Conditions: Files/folders on the Target directory will be copied to the Source Directory |
| Actors: User, System |
| Main Success Scenario:   1. User selects SyncTask and clicks on ‘Restore’ 2. System proceeds to restore the Target folder to the Source folder 3. System updates “Successful”, and last run time in the main window for the selected SyncTask   Extension(s):  2a. System encounters error during restore  2a1. System updates “Unsuccessful”, and last run time in the main window for  the selected SyncTask  Use case ends |

|  |
| --- |
| Use Case Number: 9  Use Case Name: View Source/Target Folders |
| Pre-Conditions: At least 1 SyncTask has been created |
| Post-Conditions: Source/Target folders are opened and displayed to the user using  windows explorer |
| Actors: User, System |
| Main Success Scenario:   1. User selects SyncTask and clicks on ‘Action-> Open Source/Target Folder’ 2. System opens and displays Source/Target folders in windows explorer   Extension(s):  2a. Source/Target folder does not exist.  2a1. System displays error that Source/Target path cannot be found  Use case ends |

|  |
| --- |
| Use Case Number: 10  Use Case Name: Export Synchronization Profiles |
| Pre-Conditions: At least 1 SyncTask has been created |
| Post-Conditions: All SyncTasks for profile are exported to a \*.profile file |
| Actors: User, System |
| Main Success Scenario:   1. User selects ‘Export Task’ from main menu 2. System requests from user location and filename for exported file 3. User selects location and enters filename for exported file 4. System exports all SyncTasks for current profile into location with filename selected by user |

|  |
| --- |
| Use Case Number: 11  Use Case Name: Import Synchronization Profiles |
| Pre-Conditions: A SyncProfile has been previously exported |
| Post-Conditions: SyncTasks from exported profile will be imported and added into  current profile |
| Actors: User, System |
| Main Success Scenario:   1. User selects ‘Import Task’ from main menu 2. System request from user location of file to import 3. User selects file to import 4. System imports all SyncTasks from the export file into the current profile   Extension(s):  4a. System determines that user selected file to import is not valid  4a1. System displays error message to user  Use case resumes from step 2 |

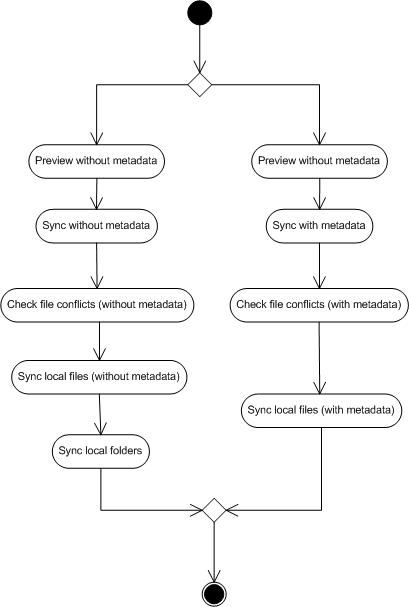
|  |
| --- |
| Use Case Number:12  Use Case Name: View Log File |
| Pre-Conditions: At least 1 SyncTask has been created |
| Post-Conditions: System displays log file to the user |
| Actors: User, System |
| Main Success Scenario:   1. User selects SyncTask and clicks on ‘Action -> View Log’ 2. System displays log file to user   Extension(s):  2a. System cannot find log file associated with selected SyncTask  2a1. System informs user that no log file exists for select SyncTask  Use case ends |

|  |
| --- |
| Use Case Number: 13  Use Case Name: View Help File |
| Pre-Conditions: - |
| Post-Conditions: Help file is displayed to user |
| Actors: User, System |
| Main Success Scenario:   1. User clicks on ‘Help’ on the main menu 2. System displays help file to user   End of User case |

Activity Diagram

Reconciler:

The following diagram is the activity diagram for reconciler.



**Activity 1: Preview without metadata**

|  |
| --- |
|  |
| Description: |

**Activity 2: Sync without metadata**

|  |
| --- |
|  |
| Description:  Retrieves local files and folders in source and destination directories respectively and performs the first-time synchronization. This activity consists of two actions namely sync local files and sync local folders. |

**Activity 3: Check files conflicts (without metadata)**

|  |
| --- |
|  |
| Description:  Checks for file conflicts by comparing every single file in source directory with its corresponding file in destination directory. Passes user settings to next block which is “Sync local files” to perform the real synchronization actions. For example, if two files have the same name but different modification time (i.e. conflict occurred); users can always choose to keep both copies, keep the latest copy, keep source copy and keep destination copy. |

**Activity 4: Sync local files (without metadata)**

|  |
| --- |
|  |
| Description:  Creates metadata of every files in both source and destination directories for this first-time synchronization. Metadata stores files’ attributes such as file name, size, last modified date and etc. This block will then perform synchronization actions which are received from “Check files conflicts” block. |

**Activity 5: Sync local folders (with metadata)**

|  |
| --- |
|  |
| Description:  Perform synchronization actions on both source and destination folders based on folders’ attributes such as folder name, size, and length. |

**Activity 6: Preview with metadata**

|  |
| --- |
|  |
| Description: |

**Activity 7: Sync with metadata (with metadata)**

|  |
| --- |
|  |
| Description:  Iterate through |

**Activity 8: Check files conflicts (with metadata)**

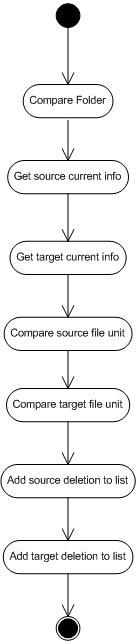
|  |
| --- |
|  |
| Description:  Iterate through the dirty source list and compare every file in the dirty source list with its corresponding file in dirty destination list to see what synchronization actions should be performed. Passes the synchronization actions to next block which is “Sync local files”. For example, if the source file has been modified and its corresponding file in destination has been deleted; then this block will pass sync action “keep source copy” to next block. |

**Activity 9: Sync local files (with metadata)**

|  |
| --- |
|  |
| Description:  Performs the real synchronization actions received from “Check files conflicts” block. For example, if the sync action received from “Check files conflicts” block is “keep source copy”; then this block will copy source file to its destination location. |

Detector

The following diagram is the activity diagram for reconciler.



**Activity 1: Compare folders**

|  |
| --- |
|  |
| Description:  Perform synchronization actions on both source and destination folders based on folders’ attributes such as folder name, size, and length. |

**Activity 2: Get current source file info**

|  |
| --- |
|  |
| Description:  Get file information (name, size, hash, last modified date, etc) for files in the source directory and store them in a list. |

**Activity 3: Get current target file info**

|  |
| --- |
|  |
| Description:  Get file information (name, size, hash, last modified date, etc) for files in the source directory and store them in a list. |

**Activity 4: Compare source file unit**

|  |
| --- |
|  |
| Description:  For each file in the source directory, compare it with source meta data. If the file was modified or newly created, add it to sDirFiles (source dirty list) and flag it as ‘M-‘ or ‘C-‘. If the file has not been modified, add it to sCleanFiles (source clean list). Lastly, delete this file unit from source meta data. |

**Activity 5: Compare target file unit**

|  |
| --- |
|  |
| Description:  For each file in the target directory, compare it with target meta data. If the file was modified or newly created, add it to tDirFiles (target dirty list) and flag it as ‘M-‘ or ‘C-‘ . If the file has not been modified, add it to tCleanFiles (target clean list). Lastly, delete this file unit from target meta data. |

**Activity 6: Add deleted file unit on source to list**

|  |
| --- |
|  |
| Description:  For each file unit left in source meta data, if the file unit is a directory, add it into sDirtyDir (source dirty directories). If the file unit is a file, add it into sFileDir (source dirty files). Flag the file unit as ‘D-‘. |

**Activity 7: Add deleted file unit on target to list**

|  |
| --- |
|  |
| Description:  For each file unit left in target meta data, if the file unit is a directory, add it into tDirtyDir (target dirty directories). If the file unit is a file, add it into tFileDir (target dirty files). Flag the file unit as ‘D-‘. |

Algorithm Description – Detector

Algorithm Description – Reconciler

|  |  |  |
| --- | --- | --- |
| Source | Target | Sync Actions |
| M | M | Keep the latest copy by comparing their last modified time. |
| D | Copy the modified file in source directory to destination directory. |
| C | Keep both copies. |
| R | Keep both copies. |
| D | M | Copy the modified file in destination directory to source directory. |
| D | No action will be performed since both copies have been deleted. |
| C | Copy the newly created file in destination directory to source directory. |
| R | Copy the renamed file in destination directory to source directory. |
| C | M | Keep both copies. |
| D | Copy the newly created file in source directory to destination directory. |
| C | Action will be taken based on user settings. |
| R | If their contents are the same; then no action will be performed. Otherwise, keep both copies. |
| R | M | Copy the modified file in destination directory to source directory. |
| D | Copy the renamed file in source directory to destination directory. |
| C | If their contents are the same; then no action will be performed. Otherwise, keep both copies. |
| R | Keep both copies. |

User settings:

Keep both copies; keep source copy; keep destination copy;

**Syncsharp Users Guide**

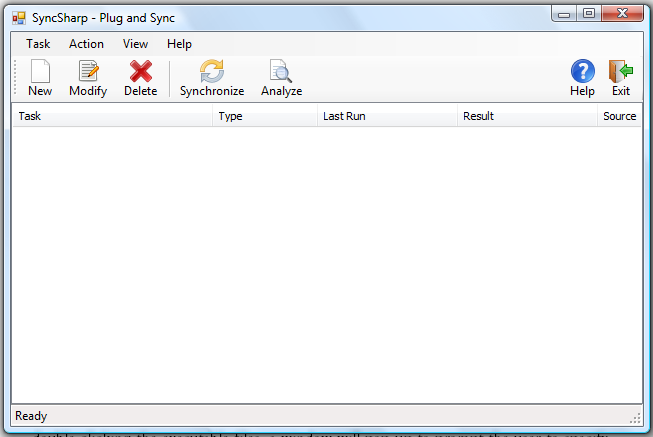
Part 1: Product Description

Syncsharp is a perfect product for tertiary students who work with multiple computers and need to synchronize files that reside in different computers through an immediate device such as a USB drive. It is a Windows based application that allows users to sync files between multiple computers through an immediate device with no installation required. The file synchronization process is streamlined through a list of pre-determined user preferences.

***Part 2:* Using Syncsharp**

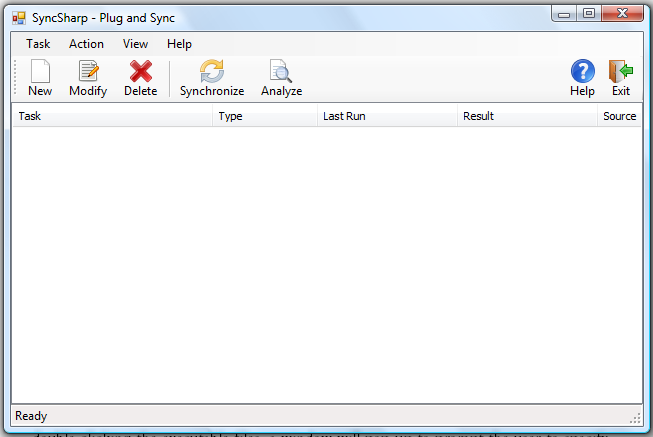
On First Run

On first run, user will need to activate Syncsharp by manually activate the program. Subsequently, the program automatically run by itself once user plug in their thumb drive or hard disk. Upon double clicking the executable files, the following window will pop up.

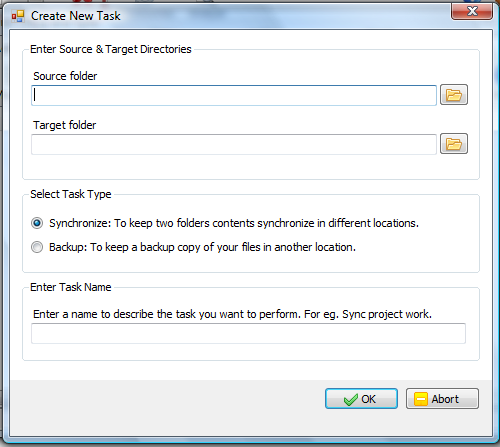


Creating/Deleting New Sync Task

Users could be able to Create, Delete or Modify Sync Tasks from the above pop up window.



Create new Sync Task 1:



Step 1: Click on New, the following window will show up.

Step 2: Click on the icon  to locate the two folders (“Source folder” and “Target folder”) that you would like to synchronize.

Step 3: Select Task Type.

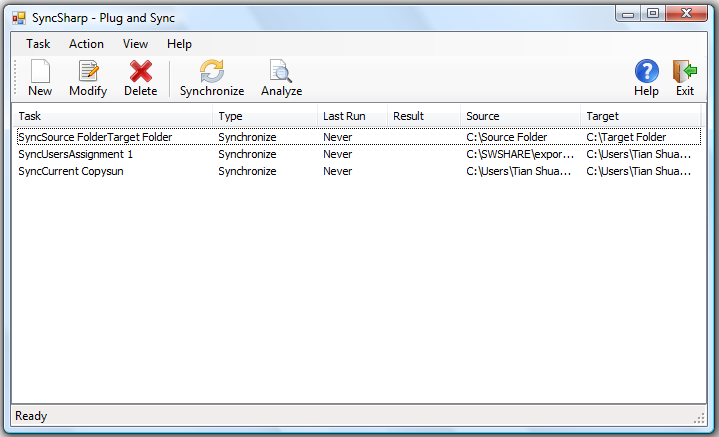
Synchronize – Synchronize source and target folders

Backup – Keep a copy of files in “Source folder” in “Target folder”.

Step 4: Specify the name of this Sync Task. The task will be saved, so next time the program is activated. It will automatically try to complete this task.

Step 5: Click on “OK” to finish this wizard.

Delete Sync Task 1:



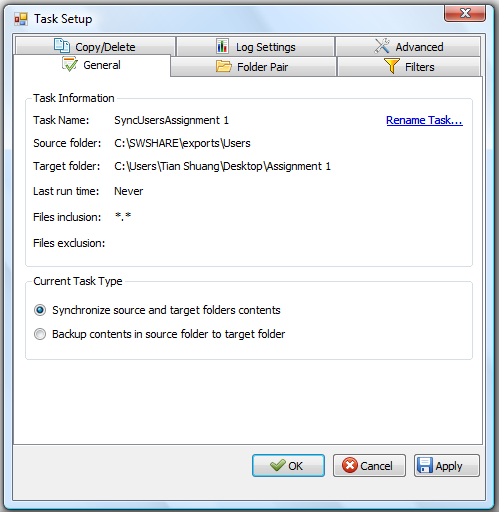
Click on the Sync Task you want to delete. 🡪 “Delete”.

Modify Sync Task

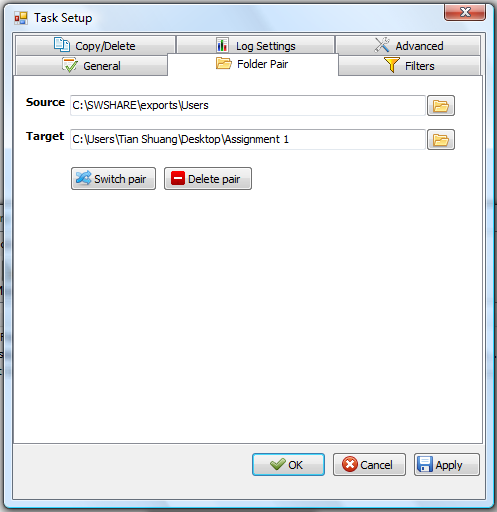
Click on the Sync Task you want to delete. 🡪 “Modify”. The “Task Setup” window will pop up. It consists of 6 tabs.

* “General” – Display general information.
* “Folder Pair” – Modify folder pairs.
* “Filter” – Filter file/folders to be exclude in syncing.
* “Copy Delete” – Modify how program handle file copy and delete.
* “Log Settings” – Modify logging options.
* “Advanced” – Advanced syncing options.

“General” Tab



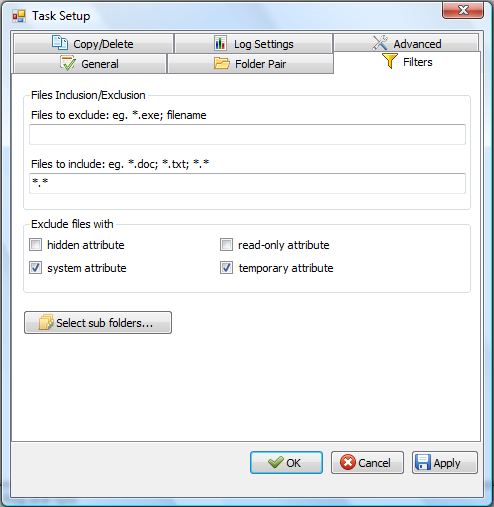
“Folder Pair” Tab



“Switch pair” – Switch the path of “Source” and “Target”

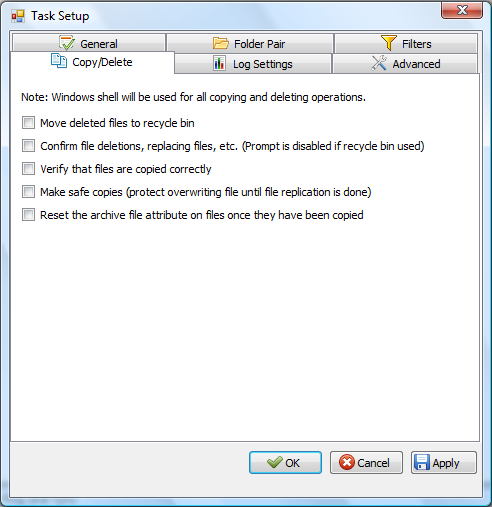
“Delete pair” – Delete the “Source” and “Target”

“Filter” Tab



“Select Sub folders…” will list out all sub folders in “Source” and “Target”. User could be able to exclude sub-folders in the syncing process.

“Copy/Delete” Tan



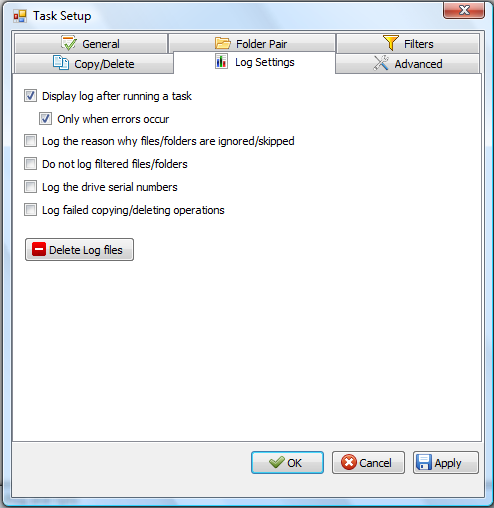
*“Move deleted files to recycle bin”* – If there is a deletion in the syncing process, the deleted file will be moved into recycle bin instead of permanently deleted from the file system.

*“Confirm file deletions, replacing files, etc.”* – If there is a deletions or replacement action in the syncing process, program will prompt user for confirmation.

*“Verify that files are copied correctly”* – If there is copy action, verify the copied files by checking file size and content.

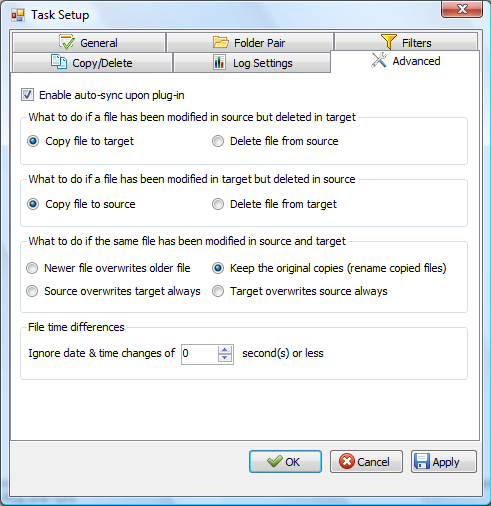
*“Make safe copies”* – While modifying or copying files, act on a temp file to protect files from corruption.

“Log setting” Tab

****

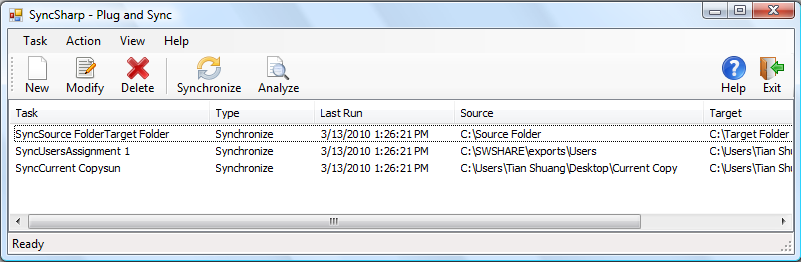
***“Delete Log files” -*** Remove the previous generated log files.

**Advanced**

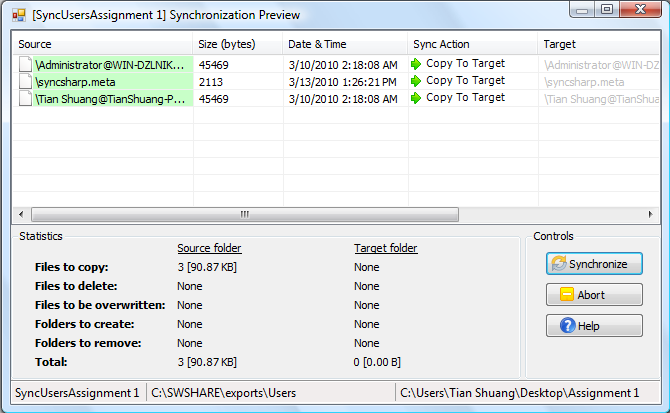
****

*“Enable auto-sync upon plug-in”* – If checked, PlugSync will be enabled. See section

Review Sync Result



On the main menu, choose which sync task you would like to review, then click on Analyze.



A window will pop up, list out all the changes going to happen. You can either click on “Synchronize” to proceed to synchronization or “About” to exit.

How to Import/Export Sync Task?

On the main menu,

Click “Task” 🡪 “Import Task …” or simply click Ctrl + I 🡪 specify which Sync Task to import.

Click “Task” 🡪 “Export Task …” or simply click Ctrl + X 🡪 specify the location you want the Sync Tasks to be saved.

PlugSync

The program will run by itself while the USB is plugged into a computer. Upon execution, the program will count down for 5 seconds waiting for user interruption. If there is no interruption, the program will automatically try to perform all the saved Sync Task.

**Glossary**

|  |  |
| --- | --- |
| Term | Definition |
| SyncSharp | Name of our sync tool |
| SyncTask | Configuration file that contains source and target directory information to be synchronized |
| SyncProfile | Contains list of SyncTasks for a Particular PC/Laptop |
| FileUnit | Abstract representation of a file or folder, contains information such as name, size, hash code, last modified date, etc. |
| PlugSync | A feature in our program. Program will run by itself while the USB is plugged into a computer. Upon execution, the program will count down for 5 seconds waiting for user interruption. If there is no interruption, the program will automatically try to perform all the saved SyncTask. |
| 1 way sync | Update destination directory to have the same content as source directory |
| 2 ways sync | Update source and destination directories to have the same state |
| Report/Logger | Log file that records the operations perform in the synchronization process |
| Target | The destination directory to be sync or compared |
| Detector | The sub-system that detect changes of the source or destination directory |
| Reconciler | The sub-system that resolves conflicts between the source & destination directories |
| TaskSettings | Contains configuration settings made for each SyncTask |

1. For current release, PlugSync feature is not fully functioning in Windows 7 due to system restrictions. [↑](#footnote-ref-1)