# OBJECTIVE

Simplify the process of moving scanned files from the scanning workstation to the editing workstation.

# PROBLEM DESCRIPTION

The existing workstations (WS) are not connected. A variety of work flows can be used to get the scanned files from the scan WS to the edit WS. Some of the issues that come up are:

1. Finding the files on the scan WS
2. Cleaning up the files on the scan WS after the files were copied (scan WS, thumb drive)
3. Putting the scanned files on the edit WS.
4. Overwriting existing files on the edit WS.
5. Copying/moving only scan files.

The current process time consuming and is comprised of many ‘steps’ making it ‘error prone’.

In addition to the obvious file management problem there is an additional hardware & security problem.

## WINDOWS XP

The scan WS requires Windows XP. This operating system presents a serious security risk and should not be connected directly to the local network or the internet (wired/Wifi). The application, requires a very specific software versions that should not be updated.

# SOLUTION

To solve this problem a combination of hardware and software is indicated.

*In any new system there will always be unexpected discoveries and limitations. Potential risks will be discussed below.*

## HARDWARE

To isolate the XP scan WS from the edit WS, a special USB networking cable will be used. The edit WS will be connected to the internet. If the edit WS is properly maintained, OS updates and anti-virus, the scan WS will be safe.

## SOFTWARE

The software solution involves two applications. The first application will run on the scan WS and will ‘push’ scanned files to the edit WS. The second application will run on the edit WS and will ‘pull’ the scanned from the edit WS. Old files will be cleaned up after they are copied.

### PUSH & PULL APPLICATIONS

### General PUSH / PULL Description

Both apps will allow the user to choose a single button which will move all files from one WS to the other WS.

The app will contain a single button. Selecting the button will start the process of moving the scan files from one WS to the other WS. The interface will also include a simple status display. A collection of menu’s will allow the user to configure all of the necessary options. The assumption is that once these options are set they will never change.

### PUSH Technical Issues

This application must be compatible with Windows XP and must run as a 32 bit application. It will have technical limitations with respect to the Microsoft libraries that may be used.

### PULL Technical Issues

This application will be a 64-bit application compatible with Windows 7 and Windows 10.

# RISKS

The USB networking cable has functionality that can’t be accessed from the PUSH and PULL applications.

The PULL application can’t access the scan WS programmatically.

Scan WS file access may be limited while the scan application is running.

# USE CASES

This section describes the variety of situations and features that apps must be able to provide for. They will form the basis for planning and development.

## Configuration

This section describes the different configuration options.

### File Types

Two file types will be supported: TIFF and JPG

One or both file types will be moved.

The file types that should be processed are set here.

### Source Folder

The source folder describes the locations were the scanned files currently exist.

This folder will be set here.

### Target Folder

The target folder describes the location were the scanned files will be copied.

This folder is set here.

### Source Tree

When files exist in a sub folder, selecting this option will copy all of the files in the folder. This feature will interact with the Target Tree configuration option.

### Target Tree

If the source tree option is configured AND files exist in a subfolder on the source system, the source tree and all files will be copied to the target folder.

## Other

### Duplicate Files

When a duplicate file or folder is discovered during the copy process the user will be warned. The user will have the option to abort all processing, cancel single file or rename the file.

Renamed files will follow the traditional format of appending (n) to the end of the file name. Example: MyPhoto\_100(2).tiff is a duplicate file name of MyPhoto\_100.tiff.

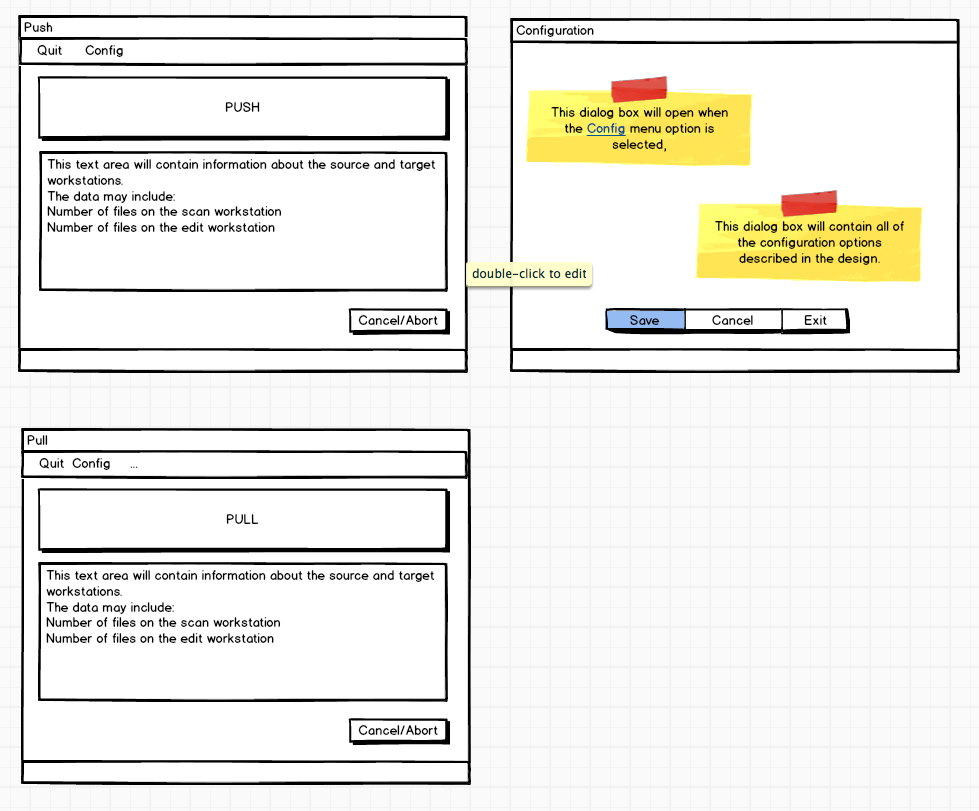
Duplicates will be tested by filename and extension. File size, create/modify dates and other file properties will be ignored.

# Mockup UI’s

PUSH = Win XP / Scan Workstation

PULL = Win 7/10 / Edit Workstation

CONFIGURATION = Common to both applications.



NOTE

All sections below this note have NOT been reviewed by the client. They have been added for documentation purposes.

# Technical Analysis

**PROJECT NAME**

|  |  |
| --- | --- |
| Win\_SourceCode/Push\_App == Solution Folder |  |
| Win\_SourceCode/Push\_App/Push ==  Project file name  Executable name | 'x' == the release version  'y' == the dev iteration  'z' == the build iteration    The starting value == Push.0.01.0 |

**PROJECT CONFIGURATION**

Use GitHub for archiving source code

**PROJECT QA**

Create MSTest project for unit testing application

* ?? Experiment ??
* .Net 2.0 and Visual Studio Pro dose not support this.
  + Alt: Try NUnit or some new test framwork

**DEVELOPMENT CHECKLIST**

Use an XML for resource strings. Do NOT use INI file.

Must be compatible with XP

Probably OK

Use for labels of target and source

Use for labels describing content

Use XHTML and SSL for displaying status info related to Target and Source.

Can I make this dynamic?

TBD

**FAIL --- EXPERIMENT**: Embed a WebBrowser control for Source/Target info.

* Create JavaScript for display/refresh
* No. Too complicated and outside the basic scope.
  + Use Windows Form controls

WORKFLOW – UX -- Add a Reset button for QA/Test

* Alt: Create a Hot Key
* Enable/disable from configuration UI

WORKFLOW -- After the Copy to target is complete:

* Cleanup source
* Prompt before cleanup
  + Do not show checkbox
  + Override/reset this from configuration UI

DATAFLOW – Verify Copy

* Use a method like CRC32 Checksum to verify copy
* This may be problematic because of the file size.

UX – Status Display

* This will be a WinForm Group
* Total counts / progress in smaller textbox
* Buttons for
  + Abort
  + Skip
  + Rename

DATAFLOW – File Types

* Modify code base to handle each file type
* Note file extension variations:
  + JPG | JPEG | etc
  + TIFF | TIF | etc

DATAFLOW – Logging

* Add logging
* Track exceptions
* Button presses
* Counts
* CRC Checksum: Pass/Fail

DATAFLOW – Rename

* Create RegEx to parse “(n)” and then increment (n+1)
* Test for duplicates based on filename and extension ONLY

UI – Status Group

* This will contain:
  + Label that displays the current file being processed
  + Label that displays “Currently Processing Record : x of y”, where x is the current number of records copies and y is the total number of records.
  + Action Buttons
    - Abort
    - Skip
    - Rename
    - ???
    - These buttons will enable/disable depending on the context.

UI – Source and Target Textboxes

* Display list of files
  + List will be dynamic. They will update as the copy progresses.
  + Scroll bar
  + Display total number of files in each folder