Funduc Software Products

Please visit our Web site at http://www.funduc.com for more information, freeware, shareware, and purchases of these and other products.

Search & Replace

Search and Replace is our award winning utility loved by programmers, webmasters, and novice computer users alike. If you need to find and replace text, or just find text, this utility is a must have.

Search and Replace looks through multiple files for a string and can also replace it with another string. It can search subdirectories and ZIP files and do case sensitive or insensitive searches. Extensive support for regular expression (egrep) search & replacement includes operations that span more than one line, insertion of the path & filename during replacements, and incrementing number replacements. Binary search & replace mode can be combined with regular expressions. Precise control over replacements is available through multiple confirmation options and a built-in context viewer that allows you to view replacements in- context, before they are made. An HTML Mode can be used to make html special character code substitutions during search/replace. Ignore Whitespace mode makes it convenient to handle spaces & cr-lf during non-regular expression operations. A built-in script editor makes frequent &/or complex multi-step search/replace operations easy to prepare. Advanced script operations include a boolean expression evaluator that provides additional control over which files are processed by the script. Advanced scripts also allow you to specify the program options in effect, insert comments in scripts, and perform iterative script operations. Multiple file masks and file date & size filters can be used to include &/or exclude files during regular and scripted operations. The internal context viewer includes a build-in editor for editing text files. The full complement of command line switches (including script launching) is ideal for network managers & advanced users. Other functions include: "Touch" files (change time/date stamp & file attributes); Preservation of original file date during replacements; Replacements on backup path files instead of the originals; Output reports to monitor operations; Shell extensions to launch Search & Replace from Explorer and the Find Menu; and more. A detailed help file with examples are included. Several foreign language interface are available from ttp://www.funduc.com.

Search and Replace has received several awards from WUGNET® (Windows User Group Net) Windows NT Shareware Pick of the Week (www.wugnet.com/shareware), Windows Magazine (www.winmag.com) Superior Shareware, and has been featured several times in Lockergnome's Free Windows 95/NT E-zine (www.lockergnome.com). It has a ZDNet Hotfiles (www.hotfiles.com) of 5 stars, & a TUCOWS (www.tucows.com) 5 Cow rating.

Search and Replace Regular Expression Wizard

Search and Replace Regular Expression Wizard is a step-by-step wizard to help you build regular expressions for use in Search and Replace. The expressions can be constructed from a menu of operators having English language counterparts. The impact of the operators is displayed in color coded form on a sample text you provide. You can move back & forth between search operators - replace operators - final results to adjust your expressions as necessary. When finished, the final expressions are inserted into the corresponding Search and Replace fields if Search and Replace is running. The Wizard itself is freeware. Documentation requires a one time purchase.

Directory Toolkit

Directory Toolkit is a multipurpose directory comparison and file manager that provides features missing from Windows. It displays single directory lists or side- by-side directory comparisons using color coding for alike, different, or missing files. Comparisons can be based on dates or a binary file analysis of file content. You can copy, move, & rename files while retaining long filenames and six built-in synchronize functions simplify updating directories. Built-in archive functions, with Explorer shell extensions, let you create & manage archives from Explorer and use Directory Toolkit as your primary archive manager for supported archives (zip, tar, gz, lz (read only), & cab). File operations and directory comparison functions work seamlessly on files in archives, e.g., no need to manually extract files first. Other features include statistics for file counts and path sizes; Ignoring filename case during comparisons; Explorer compatible OLE drag n Drop; Split/Concatenate files to/from diskettes; Encode/Decode UENCODE, Base64(Mime), BinHex(Mac), XXENCODE and yEnc formats, (Decode also includes quoted-printable, 7 bit unencoded, & plain text); An integrated comparison viewer for a color coded display of the actual differences in files; Safe physical sort of directories; Touch files (change time/date & attributes stamps); Find duplicate files on a drive; Open files and launch applications; Multiple include/exclude file masks and the ability to include/exclude files based on size and date; Command line switches & script functions to automate comparisons & synchronize operations & control program options in effect via batch files; Configurable output file and print functions to create reports of single and comparison directory lists; Detects & runs (optionally) setup programs in archives; Display file version information including 8.3 file name equivalents; File numbering to number files sequentially based on adjustable sort order and a user defined starting value. Help file with examples included. International versions are available from http://www.funduc.com.

Directory Toolkit has been a winner of the WUGNET® Windows NT Shareware Pick of the Week for (www.wugnet.com/shareware), WUGNET® Windows95 Best Choice Shareware for the week, and featured Lockergnome's Free Windows 95/NT E-zine (www.lockergnome.com). It has a ZDNet Hotfiles (www.hotfiles.com) of 5 stars.

Funduc Software ZIP Extractor

Funduc Software ZIP Extractor is ZIP "self-extractor stub" that lets you turn your ZIP files into a self-extracting executable that will automatically unZIP the archive when launched. This allows you to share your zip files with another person who might not have an unzip software on their computer &/or who might have only occasional use for an unzip tool. The self-extractor allows some or all files to be extracted and can create the paths in the zip file (default) or "flatten" the directory structure (unzip into a single path).

Registry Toolkit

Registry Toolkit lets you search &/or replace and edit the Windows registry. You can specify the data types to process, start key, case sensitive, whole word. Regular expression operations using an egrep-type syntax are also available. You can create/edit keys, values, &/or data manually if you wish. A unique 'operations' function lets you move/copy/delete/export values you select. For example, you can copy/move keys from one location in the registry to another. The program now maintains multiple 'undo' files that can be used to reverse previous changes (10 is the default). An automatic 'last change' undo is available from the program GUI. Replaces can be previewed in advance, in-context. Replaces can also be previewed in the context of an overall search - a comparison dialog then lets you select/deselect specific operations to perform before making the overall replace. The program can print search & registry browse results. The .reg import function allows you to compare existing data vs. data to be imported - specific items can then be selected/deselected. The .reg export function can be used to make special .reg files that delete keys when merged via Registry Toolkit or our new Registry Toolkit Merge program (see our web site). The various program comparison dialogs are configurable and their contents can be copied to the windows clipboard. The program can read DOS environment variables and use them in a search/replace. Script file & command line functions are available for automated &/or complex operations. Registry Toolkit will operate on a remote registry (if you have permission to do so). A German interface is available.

Registry Toolkit Merge

Registry Toolkit Merge is a companion program for Registry Toolkit. Registry Toolkit Merge is a command line that merges .reg files into the Windows registry. Registry Toolkit Merge is unique because it can be used to create self-contained, self-importing .exe's. When clicked on, the .reg information is imported into the registry (after confirmation). This is be useful, for example, if you need to distribute a .reg file to others but are worried that they might not know how to import a .reg file manually. Alternatively, a .reg file can be passed to the program from the command line. Like Registry Toolkit, the program can be used to apply .reg files that will delete keys from the registry or copy/move them to another location -- tasks that cannot be accomplished easily by the regedit.exe .reg merge function that comes with Windows.

Folder Synchronize

Folder Synchronize is a command-line utility useful to network administrators and other power users. It provides a subset of the functions in our Directory Toolkit utility, namely, the comparison & synchronize functions. We created Folder Synchronize for those who needed those functions but not the GUI interface & other functions of Directory Toolkit such as archive management, file operations, etc. Folder Synchronize is a native 32-bit executable for use with all 32 bit Windows platforms.

Folder Synchronize can synchronize two paths using one of 6 synch methods. The program can also do a comparison-only. This is useful with the output report function. When the output report is enabled it will contain a summary of the path comparison &/or a summary of the file operations carried out by the synchronize. Other features include: Switchable display (or not) of a progress meter; A custom bitmap & message of your choice can be displayed on the screen while an operation progresses; A 'No Cancel' switch can be used to make it less easy for a procedure to be interrupted; Require (or not) confirmation of file operations; Complex include/exclude file masks that let you include/exclude some file or subdirectories while including others; Multiple operations can be submitted to the program via multiple 'configuration files'.

Folder Synchronize Script Editor

The 'Script Editor' is used to generate (or edit) script files that can then be used with Folder Synchronize or our Directory Toolkit utility. The script switches are presented in an organized arrangement. The display of related switches can be collapsed/expanded for readability.

Application Mover

Application Mover is a tool that relocates installed programs from one path to another on your hard disk. Application Mover takes files found one path and moves them to another that you specify. Application Mover scans the windows registry for references to old file location and adjust the registry to the new location. Application Mover also scans all windows shortcuts in the Start Menu and adjusts those as well. *.ini and *.log files present in the program's path are updated to reflect the new location of your program.

Application Inventory

Application Inventory is a utility that scans your system for program .exe files and generates a configurable report about the files that were located. Optional follow-up scans are then initiated to cross-reference the initial scan results with windows .lnk shortcuts and/or references in the windows registry. Further details on files are available from Properties and References dialogs. Include and exclude filters are available to define the initial search by path &/or string(s) found in file version information. The fields included in the initial file scan are configurable. Results be sorted, copied to the clipboard, printed, or saved in a delimited file you specify. Application Inventory can also launch windows explorer on the path containing a file in the results list. Although Application Inventory is intended as a reporting tool, various windows properties sheets can be launched that allow you to make edits to file or shortcut properties. The program defaults to scanning for .exe files but a special switch is available to scan for another file type that you specify.

Duplicate File Finder

Duplicate File Finder is utility for 32 bit windows that locates duplicate files on in one or more paths. Unlike other utilities that only check file names, Duplicate File Finder performs a binary comparison of file contents and can locate files that are alike regardless of names. The path(s) to scan is configurable by file path and file mask(s) and filter settings can be used to include/exclude files by date, size, attribute. Results can be saved to a tab delimited file. Supplementary functions include launching files, deleting files to the recycle bin, opening Explorer on a path, and examining file properties.

File Merge Express

File Merge Express is a file comparison and merge utility for 32 bit Windows. The program displays a contents comparison of two files and can (optionally) create a third file, 'Merge File' that consists of the merge of the two files being compared. The Merge File can (optionally) be opened in the built-in text (or binary) editor. The program has options for file naming of the Merge File, including automatic naming when the two source files for the comparison have the same name. The Compare Files dialog is also used to establish parameters such line prefixes, colors, orientation of the preview, etc. Command line parameters are available to specify files to compare or a script file to use. The program can also be launched from the Windows Explorer right click menu. File Merge Express will also supplement our Directory Toolkit utility. If both programs are installed, Directory Toolkit's file contents comparison will automatically employ File Merge Express, which has more features than Directory Toolkit's internal routine.

Delivery Agent

Delivery Agent is a send-only e-mail program, with mail merge functions, for use by web site managers, network administrators, customer relations and support personnel, etc. The program is used to deliver frequently used, customer-requested e-mail messages (& MIME attachments) such as newsletters, product delivery & announcements, support notes, etc. Messages can be sent to one or more users by selecting a message from a list and entering the users' address. The program can read a text file containing a list of addresses for bulk delivery. Recipients of bulk deliveries are shielded from seeing the other recipients in the list. The program checks a POP e-mail account prior to sending message via SMTP - useful for servers that 'validate' the users current IP address prior to mail relays. Version 3.0 adds MAPI support. When activated, the program will route outgoing messages to your primary MAPI-aware e-mail program. Mail merge field name definitions, variable names, and data are stored in a single file containing the data and e-mail addresses. Variable names can be anything you like (except for the name of the e-mail address field). Mail Merge 'Constants' can also be defined. The program has a customizable header function that can be used to send web mail' or 'HTML mail'. The "To: " field can also be customized for bulk, non- mail merge deliveries. Port settings to use and maximum number of recipients per block can be configured manually. Settings and messages can be created via the program GUI or, once you are familiar with the program, via an external text editor. A full complement of command line switches allows the program to be incorporated into your .bat routines. A log file of deliveries can be enabled (or not).

Shortcut Doctor

Shortcut Doctor allows you to check your Windows links/shortcuts and see which contain targets which can no longer be found. The program can then delete the shortcuts, resolve them, or carry out search/replace operations to modify shortcut fields with or without validation of new paths. Network Drive mappings can be inserted in the Replace dialog automatically. Individual shortcuts can be edited and all shortcuts on a drive can be displayed, regardless of whether they are valid or not. Explorer Internet URLS can also be checked for problems. CD-ROM drives, network drives, and specific paths can be ignored during processing. Includes an Empty Recent Documents button. Script files, run from the command line, allow you to do multiple operations, control program options, and easily specify complex paths. A German interface is available.

Funduc Explorer Extensions

Explorer Extensions is a collection of enhancements for the Windows Explorer right-click context menu: a) Encode and Decode Email attachments - support for UUENCODE, Base64(Mime), BinHex(Mac), XXENCODE, quoted-printable, 7 bit unencoded, & plain text, b) Split and Concatenate large files into smaller ones for storage on floppies & visa versa. Split can be based on file size or a string found in a file, c) Touch files - change the time, date, & attributes of files. Touch can add (or subtract) a user specified time value from the current file time. d) File Operations - copy/move/rename/delete files using complex file masks. The program can also change the case of file names & number files. A German version is available.

Drive Bar

Drive Bar allows you to conveniently open views of your drives in the Windows Explorer. Drive Bar's Auto Hide function allows it to disappear from view when not in use so that it does not take up screen space. Drive Bar is resizable and can be set to remain visible at all times. It can also be set to launch alternative file manager programs in lieu of Windows Explorer, launch views as "Tiled" or not, ignore specified drives in the list of drive icons, and run automatically at Windows startup.

Update Toolkit

Update Toolkit is a multi-purpose application intended for administrators network managers. This program was created for those who need some or all of the separate functions of our Folder Synchronize, Shortcut Doctor, Registry Toolkit, and Search and Replace utilities under one command line application. Operations are controlled by scripts and are highly configurable. While Update Toolkit can be used on individual machines, it is oriented toward 'logon-script' operations for workstations. The program has no user interface and therefore is well suited for automated operations.

Count Characters (freeware)

Count Characters copies most Windows controls' data to the clipboard! That's just one of the features of this freeware program from Funduc Software. The program can report the type and content of some of the windows currently displayed on your desktop and for the most popular types of controls (including lists and trees) it can copy their contents to the clipboard. You can also copy the contents of password fields and regular text boxes. Cost: Free.

Decode Shell Extension (freeware)

Decode Shell Extension is very popular freeware app is an extension for the Windows Explorer. The program allows you to decode Internet email attachments by right-clicking on a saved message file from Explorer and selecting

Decode from the menu. The program supports multiple attachments, multi-part files (already concatenated), UUENCODED, Base64(MIME), BinHex (Mac format), quoted-printable, XXENCODED, plain text files, and User (table) encoded files. If the attachment contains only one file, Decode Shell Extension will offer to open it using either the associated program or Notepad. For virus safety reasons, it will not offer to launch .exe., .com, .bat, & .cmd files. You should scan decoded Word and Excel files with your anti-virus checker before loading them.

DBComp Ver 1.0 (freeware)

DBComp is a database comparison and merge application that compare two OBDC data sources for alike vs. different records and generate a "Difference File" that lists the records that either do not match or have been added/deleted from one data source compared to the other. DBComp can also perform a "Merge" operation that uses Difference File generated by DBComp and updates an OBDC data source via SQL statements. You can view the SQL statements in advance and remove one (or more) in advance if you wish. DBComp version 1.0 is freeware.

Word Search and Replace

Word Search and Replace is a freeware macro for Microsoft Word. It uses the 'find and replace' functions of MS Word to perform a search/replace in one or more files selected by a file mask supplied by the user. We created 'WordSR' for those who need to search/replace in Word documents only. The macro can use the Word 'Pattern Matching' regular expression switch. Case Sensitive and Whole Word search/replace is also possible

Excel Search and Replace

Excel Search and Replace is a counterpart to our Word Search and Replace. This is a freeware macro for Microsoft Excel that searches (and replaces) in multiple .xls files using Excel's built-in find & replace function.

PowerPoint Search and Replace

PowerPoint Search and Replace rounds out our collection of freeware find and replace Office Macros. This macro is a PowerPoint .ppt file that works with MS PowerPoint files using the find and replace functions that are part of PowerPoint. You need to have the full version of PowerPoint installed. The macro can be run as a slide show but it will not work if you only have the PowerPoint Show Viewer from Microsoft.

TZAdjust (freeware)

TZAdjust, which stands for Time Zone Adjust, is a freeware app we created for users who find it necessary to make manual changes to the Windows time zone and 'day light savings' offset parameters. TZAdjust is probably of greatest interest to Windows NT-class users. NT/2000/XP returns time information as an offset from Greenwich Mean Time (GMT). In addition, the OS automatically makes Daylight Savings compensations when reporting file times. One can disable "Automatically Adjust for Daylight Savings" entirely in the Windows Day/Time Properties dialog but there may be other instances where finer control over time zone and the daylight offset is desired. Do not use TZAdjust casually. Be sure to read the information in the 'readme.txt' before using this program.

FSHED (freeware)

FSHED, which stands for Funduc Software Hex Editor, is a freeware hex editor whose functions include File Insert; File Revert; Bookmarking; Clipboard Paste choices; Block Selection; Search/Replace; Read only switch; Print Preview; Display Choices (Bytes per line, Offset length, ANSI/OEM, Little-Endian/Big-Endian, Font Size, Screen Colors, & more). Search and Replace can be configured to use FSHED instead of HexView.

HexView for Windows (freeware)

HexView is a multiple document Hexadecimal viewer that can display, print, and print preview any file as a hex dump. It is intended as a supplement for Search and Replace for Windows, however, it is a standalone app and does not require Search and Replace. A Japanese version is available.

Limit Edit (freeware)

Limit Edit expands upon the built-in limit feature for edit boxes & rich edit controls by letting you set your own maximum size settings for the amount of text permitted in edit boxes and rich edit controls. For example, Limit Edit can set a limit on the amount of text allowed in an instance of the Notepad text editor. The Limit Edit dialog also displays interesting information about the types of controls in use by a program. Although intended for edit boxes and rich edit controls only, it may also work on some other controls that are subclasses of these. Separate 16 and 32 bit version included.

INI to Registry (freeware)

INI to Registry works well with any program that relies on .ini files. It can read the contents of any INI file and load them into the Windows registry. That way programs which previously used INI files can read the old data from the registry. The program is free. The source code is available for $25. Please E-MAIL mime@funduc.com for information on purchasing the source code.

Patch Creator

Patch Creator, one our legacy products, is a tool for software authors. It is a two part program used to make patches for updating registered versions software. Both the Patch Creator (used on the developer's end) and the Patch Applicator (applies the patch on the end user's machine) are easy to use GUI programs. Command line switches or complicated run instructions are not necessary on the user's end. The Creator and Applicator are 32 bit GUI Windows utilities. You can require passwords to run the Applicator (or not). Patches can be created to update files on users' machines from a specific version, add files regardless of existing version, remove files, or rename files. Patches that conduct 'updates' are based on binary differences between old vs. new files. Compression can be used to further reduce the size of the patch. Patches for multiple versions can be include in a single patch file. Patches can be configured to use a registry value on the end user's machine for setup path information and the applicator can also write a .reg file to the user's registry upon completion. In addition, patches can be set up to halt operations if specified files(s) are not found on the end user's machine. This can be used, for example, to not patch any files if a critical file you specify is not present. The Patch Creator has a script language for saving complex patch setups and it can be run from the command line for use in automated routines. A German version is also available.

Decode/Encode DLLs

Decode/Encode DLL is another legacy product for software developers. The Decode/Encode DLL can be called from any Windows programming language (such as VB, Visual C, Powerbuilder, etc.). It allows you to automatically detect if a given file is UUENCODED, Base64(MIME), XXENCODED, BinHex, encoded with a user-supplied table, quoted-printable, or plain text. The DLL can then decode the file(s) either in the current directory or in one specified by the user. Binary files can also be encoded for transmission over the Internet. Functions are available for encoding/decoding to/from memory buffers, listing message body text, and message headers. Utility functions for split/concatenate are also available. Example code and a model test program is included. Cost: $58 (no royalties).

Update Toolkit Help Overview

Update Toolkit is utility that provides 4 functions within a single application:

- Moving/copying files (File Operation Syntax)
- Changing and checking Windows shortcuts (Shortcut Operation Syntax)
- Searching and changing the Windows Registry (Registry Operation Syntax)
- Searching and changing text files (Text Search/Replace Operation Syntax)

Update Toolkit was created for those who need some or all of the separate functions of our Folder Synchronize, Shortcut Doctor, Registry Toolkit, and Search and Replace utilities under one command line application. For example, network managers and administrators previously had to use each application separately to make registry changes, search/replace ini files, and change .lnk's after a server migration. This can now be accomplished using a single tool, **Update Toolkit**.

Update Toolkit is launched with a single command line. Because no graphical user interface is involved, no special steps are required to disable a program GUI when routines are run on individual workstations. Progress meters and confirmations are available if you like but these can be easily disabled. Unlike some other 'program mover' type applications that give little control over the steps to be taken, **Update Toolkit** allows you to specify all aspects of operation. Furthermore, you can use any combination of the areas of functioning. For example, you can use the file copy/more/compare functions and not the other three. Or you can modify shortcuts and examine the registry but do nothing with the file or text search/replace functions.

Individual scripts are used to control each domain of functioning. A fully developed routine would use 4 separate script files. All scripts are ASCII text files that you create with your text editor. They can be any name you like.

The command line syntax is:

```
updtkt.exe "<FileOp Script>|<Shortcut Script>|<Registry Script>|<Text SR Script>"
```

where

```
<FileOp Script> = Script for file operations
<Shortcut Script> = Script for windows .lnk shortcut operations
<Registry Script> = Script for windows registry operations
<Text SR Script> = Script for text search/replace operations
```

See Command Line and 'start /w' for more information about setting up your command line.

For detailed information on the script syntax for each area, see:

File Operation Syntax
Shortcut Operation Syntax
Registry Operation Syntax
Text Search/Replace Operation Syntax

A variety of sample scripts and files listing the complete set of script switches are included in the distribution. All sample scripts are plain text files. Please review them with your text editor. The files include:

```
FileOperationScriptSyntax_Primary.txt
FileOperationScriptSyntax_Alternate.txt
ShortcutOperationScriptSyntax.txt
RegistryOperationScriptSyntax.txt
SearchReplaceScriptSyntax.txt
\SampleScripts
```

The \SampleScripts path is collection of working sample scripts and a .bat file that models how to launch scripts.

Version 2.2 Change

Users of Update Toolkit 2.1 are aware that two forms of File Operation Syntax file operations script syntax are available. Previously we referred to these as 'Directory Toolkit Syntax' and 'Folder Synchronize Syntax'.

We are moving to a unified file operations syntax based on the 'Directory Toolkit' form. We now call this 'Primary' form. The previous 'Folder Synchronize' style syntax is now referred to as 'Alternate' form.

With the exception of the function in the table below, all file operations functions are now covered in the single syntax style. If you need to use environment variables for source and target you must use the 'alternate' form.

If you are a previous user you do not need to convert any of your old scripts!!! Update Toolkit v 2.2 understands all earlier scripts.

Thank you for bearing with us while we make this transition.

Switches available in 'alternate' syntax style not yet supported in the newer, unified style:

Switch	Function
`Environment Variables for Source & Target`	`Environment variables are understood by the Source= and Target= switches.`

Users familiar with our Folder Synchronize, Shortcut Doctor, Registry Toolkit, and Search and Replace tools will recognize the similarity with those applications. **Update Toolkit** was created for those who need some or all of those separate functions of our functionalities in a single, command line application. For example, network managers and administrators previously had to use each application separately to make registry changes, search/replace ini files, and change .lnk's after a server migration. Now all this can be accomplished using a single tool, **Update Toolkit**. The program is launched with a single command line.

Important Note

Update Toolkit is intended for computer users who are familiar with the risks of making changes to the Windows Registry and Windows Shortcuts.

Backup critical data in advance. Update Toolkit does have mechanisms for backup paths and undo files. However, mistakes can cause your computer to stop operating correctly. The fix may require you to reinstall your operating system and all applications.

Use care if you are doing text search/replaces in proprietary and/or binary files such as Word .doc's, Access databases, and so on. As you know, there are internal file pointers in many proprietary files that may be disturbed if you make replacements that are not exactly the same length as the search string. If the pointers are disturbed the original app that created the file may no longer be able to read that file.

Finally, perform operations only within the intended area for a particular functionality. For example, do not use the text Search-Replace function to make changes to .lnk files or the registry.

Command Line

The **Update Toolkit** command line syntax is:

```
updtkt.exe "<FileOp Script>|<Shortcut Script>|<Registry Script>|<Text SR Script>"
```

where

```
<FileOp Script> = Script for file operations
<Shortcut Script> = Script for windows .lnk shortcut operations
<Registry Script> = Script for windows registry operations
<Text SR Script> = Script for text search/replace operations
```

If you are launching from a .bat/.cmd file we suggest you preface command lines with the windows command, start /w, if there is more than one command in your .bat/.cmd file (see below). The '`start`' launcher is available in all 32 bit windows platforms. Type '`start /?`' from your command line to get more information for your particular OS.

Notes:

- The pipe character - | - must be used to delimit the references to the script files.
- You must specify 3 | characters even if your routine does not involve all possible components (see below)
- The order of script references above cannot be changed. The program expects a file operation script first, a shortcut script second, a registry script third, and a text S-R script last.
- Operations are performed in the above order. If you need to perform a registry operation and a text change operation and then a file operation, use two separate instances of Update Toolkit. Conduct your registry and text operations during one pass and do the file operation in the second. Be sure to use `start /w` if you have both operations in a single .bat/.cmd file.
- Surround the entire collection of script file references with " characters as modeled here.
- Scripts can have any name or extension.
- Scripts can reside in any path. They can be in different paths.
- Specify a complete file path reference to any path that does not reside in the same path as updtkt.exe.
- Alternatively, you can specify a complete path for the first parameter and then leave out the path specification for the other scripts if the remaining scripts are in that same path. OR you can specify a path (ending with \) but no script for the first parameter as a way to get the program to begin using that path for the remaining scripts (see below).
- Do not surround paths with spaces with " characters. The entire script file section is already surrounded with " characters. updtkt.exe will handle spaces directly.
- An error in any single script does not abort the remaining operations.

Examples:

1. This routine uses all 4 areas:
   ```
   updtkt.exe "test.dts|test.sds|test.rts|test.srs"
   ```
2. This routine does a file operation and a text search-replace operation only:
   ```
   updtkt.exe "test.dts|||test.srs"
   ```
3. This routine does a file operation and a shortcut operation only.
   ```
   updtkt.exe "test.dts|test.sds||"
   ```
4. This routine does a registry operation only.
   ```
   updtkt.exe "||test.rts|"
   ```
5. This .bat routine does an operation using all 4 areas, followed by one using a registry operation only. Note the use of `start /w`.
   ```
   start /w updtkt.exe "test.dts|test.sds|test.rts|test.srs"
   start /w updtkt.exe "||test3.rts|"
   ```
6. Here no file operation is needed but we specify the path d:\scripts\, with the ending \ character, in that position so the program will then look in d:\scripts\ for the for the other scripts.
   ```
   updtkt.exe "d:\scripts\||reg_script.rts|sr_script.srs"
   ```

start Launcher

If you are launching instances of Update Toolkit from a .bat/.cmd file we suggest you preface command lines with the windows `start /w` command when there is more than one command in your file. The `start /w` command is used to launch commands sequentially in .bat/cmd files. The /w switch causes the routine to 'wait' for each command to finish before proceeding to the next line.

The '`start`' launcher is available in all 32 bit windows platforms. Type '`start /?`' from your command line to get more information for your particular OS.

For more information see the Microsoft Knowledge Base. Articles Q198044 and Q126410 have some information on start.exe.

Complex File Masks

All the script components of Update Toolkit accept complex file masks that can be used to include or exclude specific files &/or directories from the operation at hand. Here is some general information on how to specify complex mask that use 'exclude' specifications. The sample scripts included with the program also contain information specific to the respective areas of functioning.

Notes on Complex File Masks:

- The ~ character is the 'exclude' character.
- Complex masks are built from left -> right. In general, specify masks to include first, followed by excludes. For example, _~\images\i*.txt;*.*_ is not recommended. Use _*.*;~\images\i*.txt_ instead.
- By way of a general explanation in words:
    - **-** When a string involving the * meta character is followed by a \ character a subdirectory is referenced. For example, _*mage*\, *mages\, \image*_ refer to subdirs.
    - *\ preceding a subdir specification refers to 'any number of levels below the parent search path. For example, ~*\\_test_\ excludes any paths named _test_ any number of levels below the starting search path.
    - *\ following a subdir specification can be thought of as 'and also any subdir level below that point. For example, ~\\_test_*\ excludes \test immediately under the starting search path and all paths under \test. A path named _\whatever\test_ would be searched.
    - *\ surrounding a path means 'any number of levels below the parent search path and also everything under that. For example, ~*\\_test_*\ would exclude any path named test at any point under the parent search path. Following from the above, ~*\\_test_*\ would not search _\whatever\test_.
    - File mask specifications can be supplied after the closing \ for a subdir reference. If you don't use * in the path reference you must specify * after the \ character. For example, _*.*;~\images_* should be used and not _*.*;~\images_\ alone. In contrast, the program assumes a file spec of *.* if the * character is used in a path reference. For example, ~\image*\ is interpreted as ~\image**, which is the same as _~\image**.*_.
- Complex file masks can be used in combination with subdir references. For example, _*.*;~\test*.txt_ excludes *.txt files in \test\ but search all other files under \test
- The examples below are oriented toward excluding subdirs/files but you can use complex notation as an include. See below.
- When referencing files, * alone can be used. As most Windows files have a single . character in them this is the same as *.*
- It is OK to divide complex exclude masks into separate components separated by the ; character. For example, a mask to exclude all files named _test*_ in all subdirs and also subdir names that begin with _\test_ are best divided into separate components. See below.

Expression	**Effect**
.;~\images*	Skip files in the _images_ subdir right below the starting path for the search. You could also use *.*;~\images*.*
.;~\images*\	As above but also skips subdirs under _images_
.;~*\images*\	Skip files in any subdir named _images_
.;~*\images*.jpg	Skip .JPG files in an _images_ subdir any # of levels below the start path
.;~\images\test\	Skip files in _images\test_. *.*;~\images\test\*.* does the same thing.
.;~\images\te*\	Skip files in _images\test_, _images\testing_, etc.
.;~test*.*;~\images\te*\	Skip files named _test.txt, testing.txt, etc_ and also
\images\i*.txt	A include mask only. This searches only _i*.txt_ files in any subdir named \ _images_ any level below the search start path.
*.htm;~*_vti**\	Uses wildcards to skip all the specialized Front Page subdirs in a Front

Page web.

Sample Scripts

A variety of sample scripts and files listing the complete set of script switches are included in the distribution. All sample scripts are plain text files. Please review them with your text editor. The files include:

```
FileOperationScriptSyntax_Primary.txt
FileOperationScriptSyntax_Alternate.txt
ShortcutOperationScriptSyntax.txt
RegistryOperationScriptSyntax.txt
SearchReplaceScriptSyntax.txt
\SampleScripts
```

The \SampleScripts path is collection of working sample scripts and a .bat file that models how to launch scripts.

File Operations Syntax - Overview

Update Toolkit file operations can be used to compare two paths and/or perform synchronize operations on them. Two forms of script syntax are available for file operations: Primary and Alternate.

Please see the Version 2.2 changes topic if you used Update Toolkit v 2.1 previously for information on changes we made toward a unified syntax style for file operations. There is currently only one difference in functionality between the two syntax forms: If you need to use environment variables for the source and/or target paths you must use the Alternate syntax.

NO LONGER NEEDED

File operations can be used to compare two paths and/or perform synchronize operations on them.

There are some differences in functionality between the two forms. The table below summarizes the differences.

Functional Differences Between Script Styles

Folder Synchronize Syntax	**Directory Toolkit Syntax**
Custom BMP during operation	Not Available
Custom Message during operation	Not Available
Disable progress meter	Not Available
Environment vars for Source= & Target=	A
One operation per script file	Not Currently Available
Available	Multiple operations in a single script file
Available	Log append via the Append to Output File= switch
Not Available	Configurable output log format
Available	ZIP comments listed in output log (if enabled)
*Available	Choices about sort order of output report
	*Various filters

*Note: The two programs use different verbs for the two of the filter-related switches.
See script documentation for more info.

File Operations Syntax - Primary Syntax Form

Listed below is Update Toolkit's file operations script syntax using the 'Directory Toolkit' form. See File Operations Overview for more information.

```
# GENERAL COMMENTS
# ================

# Comment lines are prefaced with # character. Comments should
# not be placed after script commands -- keep comments on
# separate lines.

# Blank lines are ignored.

# Note: All script commands must begin in column 1!!!

# The [Default] Section, which is optional, is used to specify
# parameters for all [Action ]s in the script.  You can also specify
# parameters specific [ActionX] section(s).  For example,
# see [Action3] section below.
# All scripts must have at least one [Action ] section!! For this model
# script, [Action1] and [Default] are redundant -- both have the same
# path settings. In actual use, for single [Action] scripts you would
# not specify [Default] and would instead use [Action1] alone

# If present, [Default] should always be specified first. The order
# of processing for the [Action ] sections is determined by their number.
# They can be placed in any order within the script. Furthermore, numbers
# do not need to be inclusive -- you can skip numbers. For example, if
# your script has [Action8] [Action2] [Action5] they will be processed
# in the order: [Action2] [Action5] [Action8]. See Additional Notes for more
# information.

[Default]

# *** PATH SETTING ***
# Note: If Path1= or Path2= setting are listed in [Default], create
# a [Action1] tag with no Path1= or Path2= setting in order to use
# Path1= or Path2= as specified in [Default]. If your [Action1]
# section has a Path1= and Path2= settings they will supercede what
# is specified in [Default].
Path1=i:\test
Path2=i:\testsr

# *** SYNCHRONIZE/COMPARE OPERATION TO PERFORM ***
Operation=1
# If not specified, Default Operation = 6.
# Use only one switch per [ActionX] section.
# Use a value of 6 (Compare Only) to generate an output report
# without performing any synchronize operations.

# Note: The - value, 'prompted' switches below apply to our
# Directory Toolkit utility. The - value switches can be used to
# trigger a synchronize dialog that waits for user input. We list
# them here for those who also have our Directory Toolkit utility.
# Update Toolkit does not have a user input synchronize dialog.
# If you use a script with a - value switch Update Toolkit will
# carry out the 'No Prompt' counterpart. If you want an opportunity
# for user input with Update Toolkit, use the Confirm Updates=1
# switch.

#                   SWITCH VALUE
#                    No
# OPERATION        Prompt Prompted ACTION
# ---------------- ------ -------- ------------------
# Synchronize         0     -2     Update & Add in both directions
# Augment             1     -3     Add only new to target
# Refresh             2     -4     Update only existing in target
# Update              3     -5     Update & Add new to target
# Prune               4     -6     Update, Add new, Remove Orphans
# Remove Identical    5     -7     Remove identical files from target
# Compare Only        6     -8     Display only - No synch operations
# Recreate Dirs       7     -9     Recreate directory structure only.
#                                    No other files are copied.
# Delete Empty Dirs   8    -10     Delete empty directories from the
#                                    target. No other files are copied.
# Prompt only               -1     Display synch dialog with no default
#                                    negative values above serve as
#                                    default settings in dialog.  This
#                                    provides an opportunity to check
#                                    progress or perform other tasks
#                                    before continuing

# *** FILE MASK ***
# File Masks are specified with the Mask= switch. Complex file masks
# can be used.
# If no mask is specified the default assumed is *.*
# If you have Directory Toolkit, see the Directory Toolkit complex
# mask dialog for a builder to help you construct complex masks.
# Examples:
#  Mask                   Effect
#  --------------------   ---------------------------------------------------
#  *.i??                  Include files ending with .i followed by 3 characters
#  win.ini                This would be found by the above.
#  ~moricons.dll          Exclude this specific file.
#  ~\images\te*\          A wildcard (te*) to skip images\test, images\testing, etc.
#  ~\*\junk*\*\*.*        Skips all files in any subdir junk* and all under that
# All subdir specifications are relative Path1= and Path2=
# Separate multiple specifications with ; character. For example,
#  *.*;~\*\junk*\*\*.*
#
Mask=*.*

# *** PROCESS SUBDIRECTORIES ***
# Use the switch here to include or not include subdirectories. The
# default is 1, for search and process all subdirectories. If not
# specified, Default = 1.
# Possible Values:
#  0 = Do not recurse subdirs
#  1 = Include subdirs
Search_subdir=1

# *** OUTPUT FILE FUNCTIONS ***
# To create an Output File of the results, specify a path & file name
# using the Output File= switch. Leave blank to not write an output file
# To write to the output file in Append mode, use the
# Append to Output File=1 setting.
# IMPORTANT: Use a path for the output file other than one being used
# during a synchronize operation. If your output file is in a path that
# is being actively synchronized you may get file sharing errors.
# It is OK to define different Output Files for each [ActionX] section.
# Possible Values for append switch:
#  0 = Output file is overwritten.
#  1 = Output writes are in append mode.
Output File=
Append to Output File=1

# **** PREVIEW YOUR OPERATIONS ****
# The Destination Path= switch, if specified, is the path to copy the
# file(s) to instead of the Path2= path. For example, if your
# synchronize choice determines that letter.doc would be copied from
# the Source to the Target, if a path is specified in Destination= ,
# letter.doc would be copied there instead.
#Destination Path=c:\tools\foldsync\test_results

# **** PROGRESS METER ***
# The progress meter can be enabled/disabled with this switch. To
# make the program completely silent, disable both confirmations and
# the progress meter. The default is to display the meter.
# Possible Values:
#  0 = No meter
#  1 = Progress meter enabled
Show Progress=1

# **** DISABLE/ENABLE CANCEL BUTTON ****
# Use the values here to disable/enable the cancel button.
# The default is that cancel is permitted.
# Possible Values:
#  0 = Disable Cancel button
#  1 = Enable Cancel button
Allow Cancel=1

# **** MESSAGE TO DISPLAY ***
# Specify a brief, single line message here to be displayed
# in the progress meter.
#Custom Message=Performing a synchronize....

# **** DISPLAY CUSTOM BITMAP ***
# The program can display a bitmap (transparent) using the using the
# bottom left pixel as the transparent color.
#Bitmap File=Test1.bmp

# *** REPLACEMENT CONFIRMATIONS ***
# Controls whether a confirmation prompt for replaces is made.
# The Use OS Operations=1 switch determines the style of the
# prompting offered. If not specified, Default = 1.
# Possible Values:
#  0 = No update prompts for file replacements are supplied.
#  1 = Confirmation prompts will be made.
Confirm Updates=1

# *** PERFORM BINARY FILE CONTENTS COMPARISON ***
# Use this to perform a byte level contents comparison when two
# files have an alike name but different file date. The files will
# be considered identical if the contents are the same. Enable this
# only if you need it -- because the entire file must be read,
# processing will be slower, especially when network paths are
# involved. If not specified, Default = 0.
# Possible Values:
#  0 = Do not perform binary file checks.
#  1 = Perform binary file checks.
Check Same Size Files=0

#**** UNIX COMPARE MODES ****
# Two special 'Unix' Directory Compare modes are available as a
# subset of Binary File Comparison (above) to compare files that
# have matching date/time stamps but different file sizes. The
# first, 'Ignore Whitespace Compare', disregards whitespace
# including spaces and line terminators. This mode can be used to
# compare Unix vs. PC files that are the same name and have matching
# time/date stamps. The second, 'Ignore Case Compare', disregards
# the case of the text within the files.
# Important Notes - Please Read:
# - The Check Same Size Files=1 switch MUST also be specified for
#   either of these switches to work.
# - Ignore Whitespace Compare can be done alone.
# - To use Ignore Case Compare you MUST also enable Ignore Whitespace
#   Compare.
# - The default for both switches if not specified is =0, e.g.,
#   Ignore Whitespace Compare=0, Ignore Case Compare=0
# Settings if you want Unix Compare alone:
#  Check Same Size Files=1
#  Ignore Whitespace Compare=1
#  Ignore Case Compare=0
# Settings if you want Unix Compare and Ignore Case Compare:
#  Check Same Size Files=1
#  Ignore Whitespace Compare=1
#  Ignore Case Compare=1

# *** DISPLAY DIFFERENT / IDENTICAL FILES ***
# Use these switches to control whether the program includes
# different &/or identical in your synchronize or comparison
# operation. If not specified, Default = 1.
# Possible Values:
#  0 = Do not include those files
#  1 = Include those files
Show Different=1
Show Identical=1

# *** LOCATE SAME NAMED FILES ***
# Use the Match In Subdir=1 switch to enable a special mode to
# search for duplicate files on your disk, e.g., to scan for
# duplicate DLLs or ferret out like-named documents across
# different subdirectories. In general, you would specify the
# same path for Path1= and Path2= when using this mode, although
# other two different paths can be scanned as well. Do not enable
# Match In Subdir=1 if you do not need it - the file lists returned
# are generally not what you would want for a typical
# synchronize operation. Operation=6 is the most common usage.
# The default is Match In Subdir=0
# Possible Values:
#  0 = Do not search for duplicate files in the paths.
#  1 = Perform a Match Duplicate Files comparison.
Match In Subdir=0

# *** TIME CHECK PRECISION ***
# Because NT-class operating systems such as Windows NT/2000/XP use
# a more accurate time stamp than Windows 95/98, comparisons between
# a Win95/98 and NT-class machine may lead to false mismatches a
# second apart (Win95/98 precision is on even seconds). The
# Time Check Precision= switch can be used to establish a +/- range,
# inside of which two files will be considered identical if their
# name & size are the same.
# Specify the value in seconds to use for Time Check Precision.
# If not specified, Default = 3.
Time Check Precision=3

# *** USE OS vs. INTERNAL CONFIRMATION DIALOGS ***
# The Use OS Operations= switch controls the type of file
# confirmation dialog used: internal file vs. standard Windows
# dialogs. See Native File Operations for more information. The
# internal dialogs provide more information and choices than the
# native Windows dialogs. Some  network drivers & removable media
# may require 'native' mode. If you have trouble with subdirs not
# being created on a path, try Use OS Operations=1. If not specified,
# Default = 1.
# Possible Values:
#  0 = Use internal file confirmation dialogs.
#  1 = Use native OS file operations.
Use OS Operations=1

# *** INCLUDE FILES THAT EXIST ONLY IN ONE PATH ***
# Show Files in 1 Dir= is used to control whether orphan files in
# one or the other path are included in an operation. In most cases
# you will probably want Show Files in 1 Dir=1. If not specified,
# the default = 1.
# Possible Values:
#  0 = Do not include files that exist in one dir only.
#  1 = Include files that exist in one directory only.
Show Files in 1 Dir=1

# *** CASE SENSITIVE NAME MATCHING ***
# Use this to compare file names case sensitive or not. If not
# specified, Default = 0.
# Possible Values:
#  0 = File name matching not case sensitive.
#  1 = File name matching is case sensitive.
Case Sensitive=0

# *** OVERWRITE BLOCKS - NEWER &/or OLDER FILES ***
# The two switches below are used to prevent the replacement of
# newer and/or older files. For safety reasons you probably should
# keep 'Newer Overwrite' turned off -- If enabled, a synchronize
# could replace newer files with older versions!!! If 'Newer
# Overwrite' is not specified the program default is disabled.
# 'Older Overwrite' should probably be enabled (this is the
# default). If you disable 'Older Overwrite', older versions of
# files will not be updated with newer versions during a
# synchronize. If not specified, Newer Overwrite=0 and
# Older Overwrite= 1.
# Possible Values:
#  0 = Do not overwrite.
#  1 = Allow overwrites.
Newer Overwrite=0
Older Overwrite=1

# *** GROUPING SIMILAR FILES ***
# When Group Files=1 is specified, the program clusters files in a
# comparison display by file name, with orphaned files grouped at
# the top or bottom of the list. When Group Files=0, comparisons are
# organized such that 'missing' or 'unmatched' files are visible
# within the context of other files according to the sort order. If
# not specified, Default = 0.
# Possible Values:
#  0 = Do not group files.
#  1 = Group files.
Group Files=0

# *** DISPLAY AND OUTPUT COLUMN SETTINGS ***
# The Display Columns= and Output Columns= switches can be used to
# control the layout of Funduc Software's Directory Toolkit GUI
# display and the layout of the output report in Directory Toolkit
# and Funduc Software's Folder Synchronize. The default used for a
# script are the settings in place when the program was last run
# interactively. Describing the possible manual settings is
# complicated. Please see 'ScriptColumnDefinitionSettings.txt', that
# is installed in the program path, more information and some example
# settings.
# Display Columns=
# Output Columns=

# *** SORTING OPTIONS ***
# To sort the file listing, use one of the values below.
# To control ascending vs descending, use the Sort Type=
# switch.
# Field to sort on:
# SORT_BY_NONE = -1 (default if nothing specified)
# SORT_BY_NAME = 0
# SORT_BY_TYPE = 1
# SORT_BY_DATE = 2
# SORT_BY_SIZE = 3
# SORT_BY_PATH = 4
# SORT_BY_CHKSUM = 5
# SORT_BY_CRC32 = 6
Sort Compare=-1

# For Ascending vs Descending, Sort Type= values are:
# 0 = Descending
# 1 = Ascending
Sort Type=1

# *** FILTERING FILES ***
# The Mask= switch filters files by name &/or path.
# Other switches are available to filter files by size, date, and
# file attribute. The Reverse Filters= switch is used to cause
# these switches to function as 'exclude' or 'include'.
#
# The number of possible combinations of settings is extensive and
# difficult to explain. Please see File Operations Filter Setttings
# for more information and example settings.
# Max Size Filter=
# Min Date Filter=
# Max Date Filter=
# Skip Files Mask=0
# Ignore Attributes=55
# Reverse Filters=0

# *** ADDITIONAL ACTIONS ***
#
# Scripts can carry out more than one action. To do this, make
# [ActionX] as modeled below. They can be specified in any
# order in the script. The program parses the entire
# script and implements the 'Actions' sequentially, based
# on their number,
# e.g., [Action1], [Action2], [Action3], etc.
# Setting specified in [Default] will be in effect if that same
# switch is not specified in a given Action section.

[Action1]
Path1=i:\test
Path2=i:\testsr

[Action2]
Path1=i:\testsr
Path2=i:\test
Operation=1

[Action3]
Path1=C:\My Documents\Current Projects
Path2=C:\My Documents\Backup
Operation=1
Mask=*.doc;*.txt;~\*\Project10\*\*.doc
Confirm Updates=1
Search_subdir=1
```

File Operations Syntax - Primary Form Notes

File Operations Scripts the have two types of 'section tags' to control the order steps and various options settings. Here are some general notes about the [Default] and [Action] tags:

- The [Default] Section, which is optional, is used to specify parameters for all [Action] sections that are present in the script. For example, if you want to specify some options switches to be applied to all [Action]'s in the script you can specify them in [Default].
- Alternatively, you can omit [Default] entirely and repeat the option switches within each [Action] section.
- All scripts must have at least one [Action] section!! In general, if you have a script with only a single action, omit [Default] and just specify [Action1].
- If present, [Default] should always be specified first in the script.
- The order of processing for the [Action] sections is determined by their number. They can be placed in any order within the script. Furthermore, numbers do not need to be inclusive -- you can skip numbers. For example, if your script has [Action8] [Action2] [Action5] they will be processed in the order: [Action2] [Action5] [Action8].
- [Default] can be used to specify an Operation= to perform in the various [Action] sections. For example, the below script would do Operation=6 for [Action]s 2 and 3.

```
[Default]
Operation=6
[Action1]
Path1=C:\My Documents\Current Projects
Path2=C:\My Documents\Backup
Operation=4
[Action2]
Path1=C:\Test
Path2=D:\Test
[Action3]
Path1=C:\HTML Files
Path2=D:\HTML Files
```

File Operations Syntax - Alternate Syntax Form

Listed below is the file operations script syntax using the 'alternate' form. See File Operations Overview for more information.

GENERAL COMMENTS

```
# Comment lines in a script are prefaced with # or ' characters in
# column 1. The comment characters _must_ go in column 1. Comments
# should not be placed after script commands -- keep comments on
# separate lines from your working script switches.

# Blank lines are ignored.

# All script commands must begin in column 1!!!

# Script Syntax begins here. The [Directory Info] section header
# tag MUST be present and MUST precede the [Options] section
# heater tag.


[Directory Info]
Source=e:\funduc
Target=c:\funduc
# Notes:
# - Do not use " characters with Source= or Target=
# - Environment variables for path names are recognized. For
#   example, Source=%winbootdir% and Target=%temp% are OK.


# **** PREVIEW YOUR OPERATIONS ****
# The Destination= switch, if specified, is the path to copy the
# file(s) to instead of the Target= path. For example, if your
# synchronize choice determines that letter.doc would be copied from
# the Source to the Target, if a path is specified in Destination= ,
# letter.doc would be copied there instead.
#Destination=c:\tools\foldsync\test_results

# **** File Mask= ****
# The Default Mask is *.*
# You can specify the masks in the order that makes sense. Each mask
# overrides the previous one(s) so chose your order accordingly.
# Take care that an exclude mask does not supercede an include mask.
# Multiple Include= and Exclude= lines are permitted. Masks can
# also refer to subdir names. The syntax for this can be complex.
# Our Directory Toolkit utility has a complex builder that can be
# helpful for making model include/exclude specifications. All
# subdir specifications are relative Source= and Target=
#
# Examples:
# Mask                         Effect
# -------------------------    ------------------------------------
# Include Mask=*.i??           Include files ending with .i followed
#                                by 2 characters
# Include Mask=win.ini         This would be found by the above.
# Exclude Mask=moricons.dll    Exclude this specific file.
# Exclude Mask=\images\te*\    A wildcard (te*) to skip images\test,
#                                images\testing, etc.
# Exclude Mask=\*\junk*\*\*.* Skips all files in any subdir junk*
#                                and all under that
Include Mask=*.*
Exclude Mask=\fra\*\*.*

[Options]

# **** COMPARE/SYNCH OPERATIONS ****
# Use on the switches below to determine what type of
# operation to perform. If nothing is specified, a compare
# operation is the default.
# The choices are:
#
#   Switch                          Function
#-------------------               -----------------------------------
# Synch Option=Compare             Compares the two directories. This
#                                    is the default if nothing is
#                                    specified. A report should be used
#                                    with this function.
# Synch Option=Synch               Adds new files to both paths and
#                                    copies newer common files from
#                                    either path to the other.
# Synch Option=Augment             Adds new files from source to
#                                    target.
# Synch Option=Refresh             Updates only existing files from
#                                    source to target.
# Synch Option=Update              Update existing files from source
#                                    to target and adds new files from
#                                    source to target.
# Synch Option=Prune               First deletes files in the target
#                                    that cannot be found in the
#                                    source. It then updates existing
#                                    and adds new files from source to
#                                    target.
# Synch Option=Identical           Deletes files from the target that
#                                    are identical to those in the
#                                    source.
# Synch Option=Make Dirs           Replicates the target subdirectory
#                                    structure but does not copy files.
# Synch Option=Delete Empty        Delete empty directories from the
#                                    target. No other files are copied.
#
# NOTE ON PRUNE: Prune removes subdirs from the target only if
# there is no counterpart subdir in the target. For safety
# reasons, there must be at least one file somewhere in the
# source path for prune to remove empty subdirs. No subdirs
# will be removed from the target if no files exist in the
# source path.
Synch Option=Compare

# **** INCLUDE/EXCLUDE SUBDIRECTORIES ***
# Use the switch here to include or not include subdirectories. The
# default is 1, for search and process all subdirectories.
# Possible Values:
#  0 = Do not recurse subdirs
#  1 = Include subdirs
Search Subdirectories=1

# **** CONFIRM FILE CHANGES ***
# The program default is to provide a confirmation when replacing or
# deleting files/directories. To disable prompts and eliminate the
# opportunity to stop a replacement or deletion, use a zero value
# for Confirm Updates=. To make the program completely silent,
# disable both confirmations and the progress meter.
# Possible Values:
#  0 = No confirmations
#  1 = Confirmations
Confirm Updates=1

# **** UPDATE OLDER / NEWER FILES ***
# You can force the program to overwrite newer files by using
# the Newer Overwrite=1 setting. The defaults if not specified are
# Older Overwrite=1 and Newer Overwrite=0. Use Newer Overwrite with
# care. It will replace newer versions of files with older versions.
# Possible Values:
#  0 = Do not overwrite
#  1 = Allow overwrites
Older Overwrite=1
Newer Overwrite=0

# **** OUTPUT FILE FUNCTIONS ****
# When a file is specified with the Log File= switch a report is
# written. Use the Append to Output File= switch to write to the
# file in Overwrite vs. Append mode.
# Possible Values:
#  0 = Overwrite mode.
#  0 = Append mode.
#Log File=c:\tools\foldsync\foldsync.log
Append to Output File=0

# **** SORT ORDER OF THE OUTPUT FILE ****
# The primary sort order in the output file can be changed using the
# below values with the Sort Compare= switch. You can also change
# ascending vs descending using the Sort Type= switch.
# The Sort Type= values are:
#  Sort Type=0 - Descending
#  Sort Type=1 - Ascending
# The Sort Compare= settings are:
#  Field to sort on:
#  Sort Compare=-1 - None (default)
#  Sort Compare=0  - Name
#  Sort Compare=1  - File Type
#  Sort Compare=2  - File Date
#  Sort Compare=3  - File Size
#  Sort Compare=4  - Path
#  Sort Compare=5  - Checksum sort
#  Sort Compare=6  - CRC32 Sort
Sort Compare=-1
Sort Type=1

# **** PROGRESS METER ***
# The progress meter can be enabled/disabled with this switch. To
# make the program completely silent, disable both confirmations and
# the progress meter. The default is to display the meter.
# Possible Values:
#  0 = Progress meter enabled
#  1 = Progress meter disabled
No Progress=0

# **** ERROR REPORTING ***
# Popup messages for errors such as 'Drive Not Ready' or 'Drive Full'
# can be enabled/disabled with this switch. Regardless of the setting
# here, errors are always noted in the Output File if one is specified
# The default is to no popup error messages.
# Possible Values:
#  0 = Do not display messages
#  1 = Display error messages
Display Error Messages=1

# **** DISABLE/ENABLE CANCEL BUTTON ****
# Use the values here to disable/enable the cancel button.
# The default is that cancel is permitted.
# Possible Values:
#  0 = Disable Cancel button
#  1 = Enable Cancel button
Allow Cancel=1

# **** MESSAGE TO DISPLAY ***
# Specify a brief, single line message here to be displayed
# in the progress meter.
Message=Performing a synchronize....

# **** DISPLAY CUSTOM BITMAP ***
# The program can display a bitmap (transparent) using the using the
# bottom left pixel as the transparent color.
#Bitmap=Test1.bmp

# **** FILTERING FILES ****
# Include Mask= and Exclude Mask= switches above are used to
# filter files by name and path.
# Other switches are available to filter files by size, date, and
# file attribute. The Reverse Filters= switch is used to cause
# these switches to function as 'exclude' or 'include'.
```

```
#
# The number of possible combinations of settings is extensive and
# difficult to explain. Please see File Operations Filter Setttings
# for more information and example settings.

# The values below are the defaults if not specified (all
# files included). Omit those entirely if you do not need any
# special settings.
#Max Size Filter=
#Min Date Filter=
#Max Date Filter=
#Filter Skip Mask=0
#Ignored Attributes Mask=55
#Reverse Filters=0

# **** OUTPUT FORMATTING ****
# Formatting in the log file can be controlled with the below switch.
# The syntax is complex. Please see the file, 'DT.SFS' for more
# information.
#Output Columns=

# **** GROUP FILES ****
# To see all related files grouped (or not) in the output report.
# This switch has no effect on file operations during synchronize.
# It affects only how files are listed in the report.
# Possible Values:
#  0 = Files will not be grouped.
#  1 = Files are grouped by file name within the sort order.
#      Orphaned files are listed at the top or bottom of the list.
Group Files=0

# **** OPERATE ON IDENTICAL / DIFFERENT FILES ****
# The Show Different= and Show Identical= switches determine whether
# the program will operated on alike &/or different files. These
# determine the files that are available for pressing by all
# operations. In most cases, the default of Show Different=1 should
# be permitted! When Show Different=0, different files would not be
# synchronized. The default is to display both identical and different.
# Possible Values:
#  0 = Do not include those files
#  1 = Include those files
Show Different=1
Show Identical=1

# *** CASE SENSITIVE NAME MATCHING ***
# Use this to compare file names case sensitive or not. If not
# specified, Default = 0.
# Possible Values:
#  0 = File name matching not case sensitive.
#  1 = File name matching is case sensitive.
Case Sensitive=0

# **** PERFORM BINARY FILE CONTENTS COMPARISON ****
# When enabled, Folder Synchronize will do a binary contents
# comparison on files with matching names. Files that have alike
# contents will be considered identical even if their date/time
# stamps differ. Binary compare will probably be slower because
# files themselves must be read. Default if not specified = 0.
# Possible Values:
#  0 = Do not perform contents comparison.
#  1 = Perform binary contents comparison.
Check Same Size Files=0

# **** LIST FILES FOUND IN ONE PATH ONLY ****
# When disabled, Folder Synchronize will not consider files that do
# not have a same-named counterpart in one or the other path.
# Disable this with care - When disabled, orphan files in the source
# would not be added to the target. Conversely, they would not be
# pruned from the target. If not specified, Default = 1.
# Possible Values:
#  0 = Do not include files that exist in one dir only.
#  1 = Include files that exist in one directory only.
Show Files in 1 Dir=1


# **** UNIX COMPARE MODES ****
# Two special 'Unix' Directory Compare modes are available as a
# subset of Binary File Comparison (above) to compare files that
# have matching date/time stamps but different file sizes. The
# first, 'Ignore Whitespace Compare', disregards whitespace
# including spaces and line terminators. This mode can be used to
# compare Unix vs. PC files that are the same name and have matching
# time/date stamps. The second, 'Ignore Case Compare', disregards
# the case of the text within the files.
# Important Notes - Please Read:
# - The Check Same Size Files=1 switch MUST also be specified for
#   either of these switches to work.
# - Ignore Whitespace Compare can be done alone.
# - To use Ignore Case Compare you MUST also enable Ignore Whitespace
#   Compare.
# - The default for both switches if not specified is = 0, e.g.,
#   Ignore Whitespace Compare=0, Ignore Case Compare=0
# Settings if you want Unix Compare alone:
#  Check Same Size Files=1
#  Ignore Whitespace Compare=1
#  Ignore Case Compare=0
# Settings if you want Unix Compare and Ignore Case Compare:
#  Check Same Size Files=1
#  Ignore Whitespace Compare=1
#  Ignore Case Compare=1

# **** SCAN FOR DUPLICATE FILES ****
# This function will locate duplicate files within a single path and
# attempt to group them in the output report. Do not enable this for
# the typical comparison. Use this only with Synch Option=Compare.
# You should also enable display of idential and different files.
# If not specified, Default = 0.
# Possible Values:
#  0 = Do not search for duplicate files in the paths.
#  1 = Perform a Match Duplicate Files comparison.
Match In Subdir=0


# **** TIME CHECK PRECISION ****
# Specify the value in seconds to use for a range within which files
# will be considered to have the same time/date stamp. If not
# specified, Default = 3.
Time Check Precision=3
```

File Operations Syntax - Folder Synchronize Syntax Form

This was # FileOPSyntax_FS_Script1

Listed below is the file operations script syntax using the 'Folder Synchronize' form. See File Operations Overview for more information.

```
' GENERAL COMMENTS

'The ' character or # character in column 1 are comment characters.

'The [Directory Info] section header tag MUST be present and
'  MUST precede the [Options] section heater tag.

[Directory Info]
' *** PATHS ****
'Notes:
' - Do not use " characters with Source= or Target=
' - Environment variables for path names are recognized. For
'   example, Source=%winbootdir% and Target=%temp% are OK.
Source=e:\funduc
Target=c:\funduc

'**** PREVIEW OPERATIONS ****
'The Destination= switch, if specified, is the path to copy the
'file(s) to instead of the Target= path. For example, if your
'synchronize choice determines that letter.doc would be copied from
'the Source to the Target, if a path is specified in Destination= ,
'letter.doc would be copied there instead.
'Destination=c:\tools\foldsync\test_results

'**** File Mask= ****
' The Default Mask is *.*
'You can specify the masks in the order that makes sense. Each mask
'overrides the previous one(s) so chose your order accordingly.
'Take care that an exclude mask does not supercede an include mask.
'Multiple Include= and Exclude= lines are permitted. Masks can
'also refer to subdir names. The syntax for this can be complex.
'Our Directory Toolkit utility has a complex builder that can be
'helpful for making model include/exclude specifications. All
'subdir specifications are relative Source= and Target=
'
' Examples:
' Mask                        Effect
' --------------------------- ----------------------------------------
' Include Mask=*.i??          Include files ending with .i followed by
'                               2 characters
' Include Mask=win.ini        This would be found by the above.
' Exclude Mask=moricons.dll   Exclude this specific file.
' Exclude Mask=\images\te*\   A wildcard (te*) to skip images\test,
'                                images\testing, etc.
' Exclude Mask=\*\junk*\*\*.* Skips all files in any subdir junk* and
'                                all under that
Include Mask=*.*
Exclude Mask=\fra\*\*.*

[Options]

'**** COMPARE/SYNCH OPERATIONS ****
'Use on the switches below to determine what type of operation to
'perform. If nothing is specified, a compare operation is the default.
'The choices are:
'
' Setting                 Effect
' ---------------------   ---------------------------------------------------
' Synch Option=Compare    Compares the two directories. This is the default
'                           if nothing is specified. A report should be used
'                           with this function.
' Synch Option=Synch      Adds new files to both paths and copies newer
'                           common files from either path to the other.
' Synch Option=Augment    Adds new files from source to target.
' Synch Option=Refresh    Updates only existing files from source to target.
' Synch Option=Update     Update existing files from source to target and adds
'                           new files from source to target.
' Synch Option=Prune      This function first deletes files in the target that
                            cannot be found in the source. It then updates
'                           existing and adds new files from source to target.
' Synch Option=Identical  This function deletes files from the target that
'                           are identical to those in the source.
' Synch Option=Make Dirs  Replicates the target subdirectory structure but does
'                           not copy files.
' NOTE ON PRUNE: Prune removes subdirs from the target only if there is no
' counterpart subdir in the target. For safety reasons, there must be at least one
' file somewhere in the source path for prune to remove empty subdirs. No subdirs
' will be removed from the target if no files exist in the source path.
Synch Option=Compare

'**** INCLUDE/EXCLUDE SUBDIRECTORIES ***
'Use the switch here to include or not include subdirectories. The
'default is 1, for search and process all subdirectories.
'Search Subdirectories=0
Search Subdirectories=1

'**** CONFIRM FILE CHANGES ***
'The program default is to provide a confirmation when replacing or
'deleting files/directories. To disable prompts and eliminate the
'opportunity to stop a replacement or deletion, use a zero value
'for Confirm Updates=. To make the program completely silent,
'disable both confirmations and the progress meter.
'Confirm Updates=0
Confirm Updates=1

'**** UPDATE OLDER / NEWER FILES ***
'You can force the program to overwrite newer files by using
'the Newer Overwrite=1 setting. The defaults if not specified are
'Older Overwrite=1 and Newer Overwrite=0. Use Newer Overwrite with
'care. It will replace newer versions of files with older versions.
Older Overwrite=1
Newer Overwrite=0

'**** OUTPUT FILE FUNCTIONS ****
'When a file is specified with the Log File= switch a report is
'written. Use the Append to Output File= switch to write to the
'file in Overwrite vs. Append mode. The values are:
' Append to Output File=0 - Overwrite mode.
' Append to Output File=1 - Append mode.
'Log File=c:\tools\foldsync\foldsync.log
Append to Output File=0

'**** SORT ORDER OF THE OUTPUT FILE ****
'The primary sort order in the output file can be changed using the
'below values with the Sort Compare= switch. You can also change
'ascending vs descending using the Sort Type= switch.
'The Sort Type= values are:
' Sort Type=0 - Descending
' Sort Type=1 - Ascending
'The Sort Compare= settings are:
' Field to sort on:
' Sort Compare=-1 - None (default)
' Sort Compare=0  - Name
' Sort Compare=1  - File Type
' Sort Compare=2  - File Date
' Sort Compare=3  - File Size
' Sort Compare=4  - Path
' Sort Compare=5  - Checksum sort
' Sort Compare=6  - CRC32 Sort
Sort Compare=-1
Sort Type=1
```

```
'**** PROGRESS METER ***
'The progress meter can be enabled/disabled with this switch. To
'make the program completely silent, disable both confirmations and
'the progress meter. The default is to display the meter.
'No Progress=1
No Progress=0

'**** DISABLE/ENABLE CANCEL BUTTON ****
'Use the values here to disable/enable the cancel button.
'The default is that cancel is permitted.
'Allow Cancel=0
Allow Cancel=1

'**** MESSAGE TO DISPLAY ***
'Specify a brief, single line message here to be displayed
'in the progress meter.
Message=Performing a synchronize....

'**** DISPLAY CUSTOM BITMAP ***
'The program can display a bitmap (transparent) using the using the
'bottom left pixel as the transparent color.
'Bitmap=Test1.bmp

'**** FILTERING FILES ****
'Include Mask= and Exclude Mask= switches above are used to
'filter files by name and path.
'Other switches are available to filter files by size, date, and
'file attribute. These switches are:
'  Min Size Filter=, Max Size Filter=, Min Date Filter=,
'  Max Date Filter=, Skip Files Mask=, Ignored Attributes Mask=.
'
'The Reverse Filters= switch is used to cause these switches to
'function as 'exclude' or 'include'.
'
'The values at below for Skip Files Mask=, Ignored Attributes
'Mask=, and Reverse Filters= are the defaults if not specified (all
'files included). Omit those entirely if you do not need any
'special settings. The switches work in tandem with each other and
'the combinations of values is very complex. Please see the file,
''FS_ScriptFilterSettings.txt', that is installed with the program
'for more information.

'Please note that Directory Toolkit uses different verbs for Filter
'Skip Mask= and Ignored Attributes Mask=. In Directory Toolkit
'syntax the counterparts are Skip Files Mask= and Ignore
'Attributes=.
'
'Max Size Filter=
'Min Date Filter=
'Max Date Filter=
'Filter Skip Mask=0
'Ignored Attributes Mask=55
'Reverse Filters=0

'**** OUTPUT FORMATTING ****
'Formatting in the log file can be controlled with the below switch.
'The syntax is complex. Please see the file, 'DT.SFS' for more
'information.
'Output Columns=

'**** GROUP FILES ****
'To see all related files grouped (or not) in the output report.
'This switch has no effect on file operations during synchronize.
'It affects only how files are listed in the report.
' Group Files=0 - Files will not be grouped.
' Group Files=1 - Files are grouped by file name within the sort
'                 order. Orphaned files are listed at the top or
'                 bottom of the list.
Group Files=0
```

```
'**** OPERATE ON IDENTICAL / DIFFERENT FILES ****
'The Show Different= and Show Identical= switches determine whether
'the program will operated on alike &/or different files. These
'determine the files that are available for pressing by all
'operations. In most cases, the default of Show Different=1 should
'be permitted! When Show Different=0, different files would not be
'synchronized. The values are
' Show Different=0 - Do not include different files
' Show Different=1 - Include files that differ (default)
' Show Identical=0 - Do not include identical files
' Show Identical=1 - Include files that are identical (default)
Show Different=1
Show Identical=1

'**** PERFORM BINARY FILE CONTENTS COMPARISON ****
'When enabled, Folder Synchronize will do a binary contents
'comparison on files with matching names. Files that have alike
'contents will be considered identical even if their date/time
'stamps differ. Binary compare will probably be slower because
'files themselves must be read. Default if not specified = 0.
' Check Same Size Files=0 - Do not perform contents comparison.
' Check Same Size Files=1 - Perform binary contents comparison.
Check Same Size Files=0

'**** SCAN FOR DUPLICATE FILES ****
'This function will locate duplicate files within a single path and
'attempt to group them in the output report. Do not enable this for
'the typical comparison. Use this only with Synch Option=Compare.
'If not specified, Default if not specified = 0.
' Match In Subdir=0 - Do not search for duplicate files in the paths.
' Match In Subdir=1 - Perform a Match Duplicate Files comparison.
Match In Subdir=0

'**** TIME CHECK PRECISION ****
'Specify the value in seconds to use for a range within which files
'will be considered to have the same time/date stamp. If not
'specified, Default = 3.
Time Check Precision=3

'**** LIST FILES FOUND IN ONE PATH ONLY ****
'When disabled, Folder Synchronize will not consider files that do
'not have a same-named counterpart in one or the other path.
'Disable this with care - When disabled, orphan files in the source
'would not be added to the target. Conversely, they would not be
'pruned from the target. If not specified, Default = 1.
' Show Files in 1 Dir=0 - Do not display (& include in the Output
'                         File) files that exist in one dir only.
' Show Files in 1 Dir=1 - Display (& include) files that exist in one
'                         directory only.
Show Files in 1 Dir=1
```

File Operations Syntax - Directory Toolkit Syntax Form

This was

# FileOPSyntax_DT_Script1

K File Operations Syntax;File Operations Syntax:Directory Toolkit Style

Listed below is Update Toolkit's file operations script syntax using the 'Directory Toolkit' form. See File Operations Overview for more information.

```
# GENERAL COMMENTS
#
# Comment lines are prefaced with # character.
#
# The [Default] Section, which is optional, is used to specify
# parameters for all [Action ]s in the script.  You can also specify
# parameters specific [ActionX] section(s).  For example,
# see [Action3] section below.
# All scripts must have at least one [Action ] section!! For this model
# script, [Action1] and [Default] are redundant -- both have the same
# path settings. In actual use, for single [Action] scripts you would
# not specify [Default] and would instead use [Action1] alone

[Default]

# *** PATH SETTING ***
# Note: If Path1= or Path2= setting are listed in [Default], create a
# [Action1] tag with no Path1= or Path2= setting in order to use
# Path1= or Path2= as specified in [Default].
# If your [Action1] section has a Path1= and Path2= settings
# they will supercede what is specified in [Default].
Path1=i:\test
Path2=i:\testsr

# *** SYNCHRONIZE/COMPARE OPERATION TO PERFORM ***
# If not specified, Default Operation = 6.
# The synchronize operation to use is established using the switches below.
# Use only one switch per [ActionX] section.
# However, different [ActionX] sections within the same script
# file may use different operations.
# Use a value of 6 (Compare Only) to generate an output report
# without  performing any synchronize operations.

# OPERATION        SWITCH VALUE     ACTION
# ---------------- --------------  ------------------
# Synchronize      0 (-2 prompted) Update & Add in both directions
# Augment          1 (-3 prompted) Add only new to target
# Refresh          2 (-4 prompted) Update only existing in target
# Update           3 (-5 prompted) Update & Add new to target
# Prune            4 (-6 prompted) Update, Add new, Remove Orphans
# Remove Identical 5 (-7 prompted) Remove identical files from target
# Compare Only     6 (-8 prompted) Display only - No synch operations
# Recreate Dirs    7               Recreate directory structure only - No
#                                   files are copied.
# Prompt only      -1              Display synch dialog with no default
#                                   negative values above serve as default
#                                   settings in dialog.  This can give you
#                                   the opportunity to check progress or
#                                   perform other cleanup before continuing
Operation=1

# *** FILE MASK ***
# File Masks are specified with the Mask= switch. Complex file masks
# can be used.
# If no mask is specified the default assumed is *.*
# If you have Directory Toolkit, see the Directory Toolkit complex
# mask dialog for a builder to help you construct complex masks.
# Examples:
#  Mask                     Effect
#  --------------------     ---------------------------------------------------
#  *.i??                    Include files ending with .i followed by 3 characters
#  win.ini                  This would be found by the above.
#  ~moricons.dll            Exclude this specific file.
#  ~\images\te*\            A wildcard (te*) to skip images\test, images\testing, etc.
#  ~\*\junk*\*\*.*          Skips all files in any subdir junk* and all under that
# All subdir specifications are relative Path1= and Path2=
# Separate multiple specifications with ; character. For example,
#  *.*;~\*\junk*\*\*.*
#
Mask=*.*

# *** PROCESS SUBDIRECTORIES ***
# 0 = Do not search subdirectories.
# 1 = Search subdirectories.
# If not specified, Default = 1.
Search_subdir=1

# *** OUTPUT FILE FUNCTIONS ***
# To create an Output File of the results, specify a path & file name
# using the Output File= switch. Leave blank to not write an output file
# To write to the output file in Append mode, use the
# Append to Output File=1 setting.
# 0 = No. The Output File specified in [Default] is overwritten by
#     each subsequent [ActionX] operation.
# 1 = The Output File specified in [Default] is written to in Append
#     mode. Subsequent [ActionX] operations are appended. Likewise, if
#     the file already exists, any existing content is preserved.
# It is OK to define different Output Files for each [ActionX] section.
Output File=
Append to Output File=1

# *** REPLACEMENT CONFIRMATIONS ***
# Controls whether a confirmation prompt is made. The Use OS Operations=1
# switch determines the style of the prompting offered.
# 0 = No confirmation file update prompts for file replacements are
#     supplied.
# 1 = Confirmation prompts will be made.
# If not specified, Default = 1.
Confirm Updates=1

# *** PERFORM BINARY FILE CONTENTS COMPARISON ***
# Use this to perform a byte level contents comparison when two
# files have an alike name but different file date. The files will
# be considered identical if the contents are the same. Enable this
# only if you need it -- because the entire file must be read,
# processing will be slower, especially when network paths are
# involved.
# Values:
# 0 = Do not perform binary file checks on liked named files of the
#     same size.
# 1 = Perform binary file checks.
# If not specified, Default = 0.
Check Same Size Files=0

# *** DISPLAY DIFFERENT / IDENTICAL FILES ***
# Use these switches to control whether the program includes
# different &/or identical in your synchronize or comparison
# operation.
# Values:
# 0 = Do not display & include in the Output File (if specified)
#     the files that are different/identical.
# 1 = Display (& include) files that are different/identical.
# If not specified, Default = 1.
Show Different=1
Show Identical=1

# *** LOCATE SAME NAMED FILES ***
# Use the Match In Subdir=1 switch to enable a special mode to
# search for duplicate files on your disk, e.g., to scan for
# duplicate DLLs or ferret out like-named documents across
# different subdirectories. In general, you would specify the
# same path for Path1= and Path2= when using this mode, although
# other two different paths can be scanned as well. Do not enable
# Match In Subdir=1 if you do not need it - the file lists returned
# are generally not what you would want for a typical
# synchronize operation. Operation=6 is the most common usage.
# Values:
# 0 = Do not search for duplicate files in the paths.
# 1 = Perform a Match Duplicate Files comparison. When 1 is
#     specified here double check that you specified identical
#     paths for Path1 & Path2.
# If not specified, Default = 0.
Match In Subdir=0

# *** TIME CHECK PRECISION ***
# Because Windows NT/2000 uses a more accurate time stamp for
# files than Windows 95/98, directory comparisons across a network
# between a Win95/98 machine and a machine running NT may lead to
# false mismatches a second apart (Win95/98 precision is on even
# seconds). Time Check Precision can be used to establish a +/- range,
# inside of which two files will be considered identical if their
# name & size is the same.
# Specify the value in seconds to use for Time Check Precision.
# If not specified, Default = 3.
Time Check Precision=3

# *** USE OS vs INTERNAL CONFIRMATION DIALOGS ***
# The Use OS Operations= switch controls internal file
# confirmation dialogs instead of standard Windows 95/NT dialogs.
# Native file operations dialogs provide more information and choices
# than the native Windows dialogs. Please note however that some
# network drivers & removable media may require 'native' mode. If
# you have trouble with subdirs not being created on a path, try
# Use OS Operations=1
# Values:
# 0 = Use Directory Toolkit's internal file confirmation dialogs.
# 1 = Use native file operations.
# If not specified, Default = 1.
Use OS Operations=1

# *** INCLUDE FILES THAT EXIST ONLY IN ONE PATH ***
# Show Files in 1 Dir= is used to control whether orphan files in
# one path or the other are included in an operation.
# In most cases  you will probably want Show Files in 1 Dir=1.
# Values:
# 0 = Do not display (& include in the Output File) files that exist
#     in one directory only.
# 1 = Display (& include) files that exist in one directory only.
# If not specified, Default = 1.
Show Files in 1 Dir=1

# *** CASE SENSITIVE NAME MATCHING ***
# Use this to compare names case sensitive or not.
# Values:
# 0 = Do not consider the file name case when determining if two like
#     named files are different.
# 1 = File name matching will be case sensitive.
# If not specified, Default = 0.
Case Sensitive=0

# *** OVERWRITE BLOCKS - NEWER &/or OLDER FILES ***
# The two switches below are used to prevent the overwrite of
# newer and/or older files. For safety reasons you probably should
# keep 'Newer Overwrite' turned off -- If enabled, a synchronize
# could replace newer files with older versions!!! If 'Newer
# Overwrite' is not specified the program default is disabled.
#
# 'Older Overwrite' should probably be enabled (this is the default).
# If you disable 'Older Overwrite', older versions of files will not
# be updated with newer versions during a synchronize.
# Values:
# 0 = Do not overwrite files (or offer confirmation prompts) during
#     any of the synchronize operations that "update" existing files.
# 1 = Permission granted to overwrite files.
# If not specified, Newer Overwrite=0 and Older Overwrite= 1.
Newer Overwrite=0
Older Overwrite=1

# *** DISPLAY AND OUTPUT COLUMN SETTINGS ***
# The Output Columns= switches can be used to control the layout of the
# output file. Please see 'ScriptColumnDefinitionSettings.txt', that
# is installed in the program path, more information and some example
# settings.
# Output Columns=

# *** GROUPING SIMILAR FILES ***
# When Group Files=1 is specified, the program clusters files in a
# comparison display by file name, with orphaned files grouped at
# the top or bottom of the list. When Group Files=0, comparisons are
# organized such that 'missing' or 'unmatched' files are visible
# within the context of other files according to the sort order.
# The values are:
# 0 = Do not group files.
# 1 = Group files.
# If not specified, Default = 0.
Group Files=0

# *** SORTING OPTIONS ***
# To sort the file listing, use one of the values below.
# To control ascending vs descending, use the Sort Type=
# switch.
# Field to sort on:
# SORT_BY_NONE = -1 (default if nothing specified)
# SORT_BY_NAME = 0
# SORT_BY_TYPE = 1
# SORT_BY_DATE = 2
# SORT_BY_SIZE = 3
# SORT_BY_PATH = 4
# SORT_BY_CHKSUM = 5
# SORT_BY_CRC32 = 6
Sort Compare=-1

# For Ascending vs Descending, Sort Type= values are:
# 0 = Descending
# 1 = Ascending
Sort Type=1

# *** FILTERING FILES ***
#
# The Mask= switch filters files by name &/or path.
#
# Other switches are available to filter files by size, date, and
# file attribute. These switches are:
#  Min Size Filter=, Max Size Filter=, Min Date Filter=,
#  Max Date Filter=, Skip Files Mask=, and Ignore Attributes=.
#
# Reverse Filters= switch is used to cause these switches to
# function as 'exclude' or 'include'.
#
# The number of possible combinations of settings is extensive and
# difficult to explain. Please see File Operations Filter Setttings or the file,
# ScriptFilterSettings.txt, that is installed in the program path
# more information and some example settings.
#
# Max Size Filter=
# Min Date Filter=
# Max Date Filter=
# Skip Files Mask=
# Ignore Attributes=
# Reverse Filters=
# *** ADDITIONAL ACTIONS ***
#
# Scripts can carry out more than one action. To do this, make
# [ActionX] as modeled below. They can be specified in any
# order in the script. The program parses the entire
# script and implements the 'Actions' sequentially, based
# on their number,
# e.g., [Action1], [Action2], [Action3], etc.
# Setting specified in [Default] will be in effect if that same
# switch is not specified in a given Action section.

[Action1]
Path1=i:\test
Path2=i:\testsr

[Action2]
Path1=i:\testsr
Path2=i:\test
Operation=1

[Action3]
Path1=C:\My Documents\Current Projects
Path2=C:\My Documents\Backup
Operation=1
```

File Operations - Filter Skip Settings

The file skip, attribute, and reverse filters switches in a File Operations script can be employed to specify files to exclude by file attribute. This includes subdirectory entries.

Please note that attribute excludes may create situations where a file that has a match by name/date may be seen as a match by the program with a counterpart file if the two files have different attribute stamps. If you are unsure about your file sets you may wish to keep confirmations turned on and/or prepare a comparison report in advance of a synchronize operation.

Please note that there is a difference in the switch verbs for the Primary vs. Alternate syntax forms:

Primary Syntax Form	**Alternate Syntax Form**
Skip Files Mask=	Filter Skip Mask=
Ignore Attributes=	Ignored Attributes Mask=

The file skip, attribute, and reverse filter switches work in combination with each other. The number of combinations of values is extensive. Some example settings are below. Additional settings are possible. Write support@funduc.com if you need help.;-)

```
# *** SIZE & DATE EXCLUDE FILTERS ***
# Size & date filters exclude files by size &/or date. Any files
# that are less than the minimum size (in bytes) will be excluded
# from the operation. If you use Reverse Filters=1, the settings
# work as 'include' filters. Dates use the format specified in
# Control Panel (Regional Settings). Leave blank or omit entirely if
# you want no setting.
# Min Size Filter=0
# Max Size Filter=100000
# Min Date Filter=8/21/1998
# Max Date Filter=8/23/1998

# *** REVERSE FILTERS ***
# Reverse filters= is used to make the other filter switches
# function as exclude or include filters. For example, Reverse
# Filters=0 & Max Date Filter=01/01/2000 excludes all files with a
# stamp after Jan 01, 2000. Reverse Filters=1 changes this to
# include only files dated Jan 01, 2000 or after.
# The values are:
# 0 = Do not use Reverse Filtering.
# 1 = Reverse Filter mode enabled.
# If not specified, Default = 0.
Reverse Filters=0

# *** FILE & DIRECTORY ATTRIBUTE FILTERS ****
# The Skip Files Mask= and Ignore Attributes= switches are used
# together, in conjunction with Reverse Filters= to exclude/include
# some file types in the operation.
# The default values of Skip Files Mask=0 & Ignore Attributes=55
# mean no files or subdirs are filtered. If you wish to use these
# defaults it is not necessary to specify these switches.

# Please use the below list for some common settings.

# *** SOME SAMPLE SETTINGS ****
# Paste all three script switches into your script.

# Include files that are Read-Only along with any other attribute
Skip Files Mask=1
Ignore Attributes=54
Reverse Filters=1

# Exclude files that are Read-Only along with any other attribute
Skip Files Mask=1
Ignore Attributes=54
Reverse Filters=0
```

```
# Include files that are solely Read-Only
Skip Files Mask=1
Ignore Attributes=16
Reverse Filters=1

# Exclude files that are solely Read-Only
Skip Files Mask=1
Ignore Attributes=16
Reverse Filters=0

# Include Archive along with any other attribute
Skip Files Mask=32
Ignore Attributes=23
Reverse Filters=1

# Exclude Archive along with any other attribute
Skip Files Mask=32
Ignore Attributes=23
Reverse Filters=0

# Include solely Archive
Skip Files Mask=32
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Archive
Skip Files Mask=32
Ignore Attributes=16
Reverse Filters=0

# Include System along with any other attribute
Skip Files Mask=4
Ignore Attributes=51
Reverse Filters=1

# Exclude System along with any other attribute
Skip Files Mask=4
Ignore Attributes=51
Reverse Filters=0

# Include solely System
Skip Files Mask=4
Ignore Attributes=16
Reverse Filters=1

# Exclude solely System
Skip Files Mask=4
Ignore Attributes=16
Reverse Filters=0

# Include Hidden along with any other attribute
Skip Files Mask=2
Ignore Attributes=53
Reverse Filters=1

# Exclude Hidden along with any other attribute
Skip Files Mask=2
Ignore Attributes=53
Reverse Filters=0

# Include solely Hidden
Skip Files Mask=2
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Hidden
Skip Files Mask=2
Ignore Attributes=16
Reverse Filters=0

# Include Read-Only & Hidden along with any other attribute
```

```
Skip Files Mask=3
Ignore Attributes=52
Reverse Filters=1

# Exclude Read-Only & Hidden along with any other attribute
Skip Files Mask=3
Ignore Attributes=52
Reverse Filters=0

# Include solely Read-Only & Hidden Alone
Skip Files Mask=3
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Read-Only & Hidden
Skip Files Mask=3
Ignore Attributes=16
Reverse Filters=0

# Include Read-Only, Hidden, & System along with any other attribute
Skip Files Mask=7
Ignore Attributes=48
Reverse Filters=1

# Exclude Read-Only, Hidden, & System along with any other attribute
Skip Files Mask=7
Ignore Attributes=48
Reverse Filters=0

# Include solely Read-Only, Hidden, & System
Skip Files Mask=7
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Read-Only, Hidden, & System
Skip Files Mask=7
Ignore Attributes=16
Reverse Filters=0

# Include files with No Attribute Setting
Skip Files Mask=0
Ignore Attributes=16
Reverse Filters=1

# Exclude files with No Attribute Setting
Skip Files Mask=0
Ignore Attributes=16
Reverse Filters=0
```

File Operations - Find Duplicate Files

Update Toolkit has a special File Operations function to locate duplicate files on your hard disk. This function allows you, for example, to scan for duplicate DLLs or ferret out like-named documents across different subdirectories. You can file duplicate files by matching on file names or CRC32/Checksum values.

File Operations - Time Check Precision

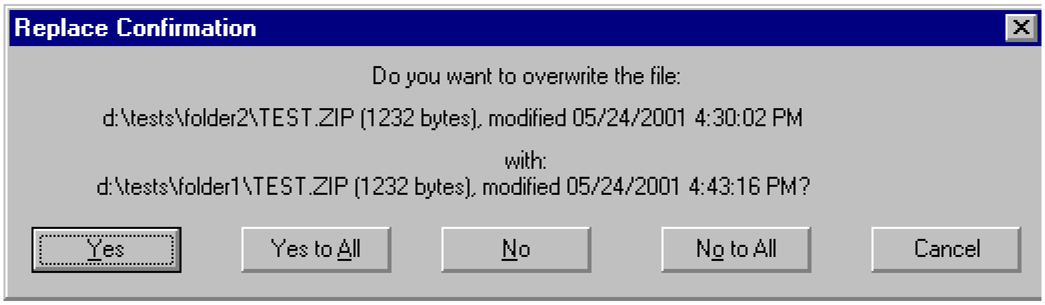
Because Windows NT/2000 uses a more accurate time stamp for files than Windows 95/98, directory comparisons across a network between a Win95/98 machine and a machine running NT may lead to false mismatches a second apart (Win95/98 precision is on even seconds). You can set the time check precision in your File Operations script thus avoiding this inconsistency between the two operating systems. A setting of 3 seconds is a fairly safe setting for most general uses.

File Operations - Native File Operations

When Native File Operations is disabled in a File Operations script, Update Toolkit uses an internal file confirmation dialog instead of standard ('native') Windows 95/NT dialogs. e Windows dialogs. When Native File Operations are off, file deletions do not use the Recycle bin (you'll still be prompted). In addition, errors on a single file will not cause the entire operation to abort.

File Operations - Internal Confirmation Prompt Dialog

The internal confirmation prompt dialog for File Operations using when confirmation prompts are enabled and internal file operations (Use OS Operations switch) is disabled:



Replace Confirmation

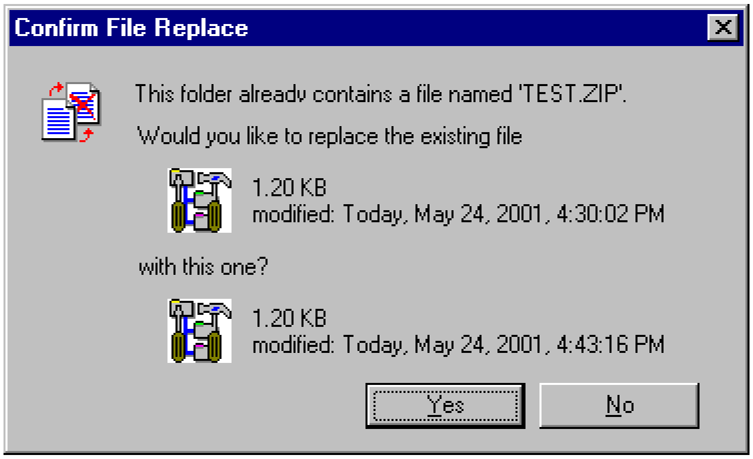
Do you want to overwrite the file:

d:\tests\folder2\TEST.ZIP (1232 bytes), modified 05/24/2001 4:30:02 PM

with:
d:\tests\folder1\TEST.ZIP (1232 bytes), modified 05/24/2001 4:43:16 PM?

File Operations - Native Confirmation Prompt Dialog

The 'native' OS file change confirmation prompt dialog for File Operations when the `Use OS Operations=1` switch is used. This contrasts with the internal dialog presented when `Use OS Operations=0` is specified.



This folder already contains a file named 'TEST.ZIP'.

Would you like to replace the existing file

1.20 KB
modified: Today, May 24, 2001, 4:30:02 PM

with this one?

1.20 KB
modified: Today, May 24, 2001, 4:43:16 PM

Shortcut Operations Syntax

Listed below is the script syntax for Update Toolkit operations on windows .lnk files. Please note that operations ALWAYS replace. The program defaults to offering replacement prompts but specify `Prompt On Replace=1` for testing.

```
# GENERAL COMMENTS

# Comment lines are prefaced with the # character.

# Script sections are demarked with bracketed - [ActionX] -
# titles, where X are numbers demarking the sequence that
# the Actions should be performed in. The can be specified
# in any order in the script file itself - the program parses
# the entire script and performs [Action2] before [Action3]
# even though [Action2] might be written last in the file.
# The [Default] section (optional) should always be the first
# item in the script. [Default] is used to set program
# options for all [Action]'s. Individual [Action]'s can
# have their own options that in turn override [Default]
# for that action.

# IMPORTANT NOTE
# ==============
# At least one [Action] must be present even if [Default] is
# present. [Default] by itself will do nothing. If your script
# has only one Action, we suggest you not specify [Default] and
# instead supply all your parameters under [Action1].
# At minimum, a path must be specified for each [Action]. The
# other parameters are optional.
# ==============================================================

# How to perform a "Check-Only" without searching for a specific
# string or making a replace?
# To perform a check-only, omit the Search_string= and
# Replace_string= switches for that [Action]

# Pathnames with spaces should NOT be surrounded by quote
# characters.

# If your script has multiple actions and if you are using output
# file(s), use a different Output_File= specification for each
# [Action] section. File writes to the output file are 'overwrite',
# not 'append'.

[Default]

# *** SEARCH PATH ***
# Use Path= to specify the path to operate on. Do
# not surround paths with spaces with quote characters.
# Scripts can also use the string,
#    Path=Local Hard Drives
# to search all local hard drives on the active machine.
Path=c:\winnt

# *** FILE MASK ***
# The default is Mask=*.lnk;*.url
# More masks are possible. Use the ~ character to exlude.
# For example:
# .lnk;*.url;~\utilities\     Skip shortcuts located C:\utilities
# *.lnk;*.url;~\utilities\*\  As above but also skips subdirs under
#                               C:\utilities
# *.lnk;~\*\utlities\*\       Skips shortcuts in any subdir named
#                               utilities & any shortcuts located in
#                               any subdir under a subdir named
#                               utilities. No internet shortcuts will
#                               be processed because *.url is not
#                               specified
# s*.lnk;~\recycled\          Shortcuts beginning with 's' and ignore
#                               the recycled bin.
Mask=*.lnk;*.url

# *** IGNORE TARGET PATHS ***
# To skip shortcuts based the path in their "target" field,
# specify one or more paths in the Ignore Drives= switch. Separate
# multiple path with the pipe character |. For example:
# Ignore Drives=f:\|d:\work\|q:\path\path2

#*** SEARCH / REPLACE STRINGS ***
# Use the switches below to specify  the search and replace strings.
#   Do NOT surround strings w/ spaces with quotes. The program does not
#   do search-only operations so use care if you use a blank replace
#   string. Consider specifying Prompt On Replace=1 for your testing
#   to give you control of replaces.
# To search for multiple strings within a single pass, separate
#   strings using the | character. The number of strings must be balanced.
#   There is no way to literalize the | character so if you need to
#   search/replace for that character, place that operation in a separate
#   action. Example of using |:
#   Search_string=c:\Windows|Testing
#   Replace_string=d:\WindowsNew|TestingOld

Search_string=c:\windows
Replace_string=c:\windows

#*** OUTPUT FILE ***
# If Output_File is blank or not set in Program
# Options from the last time the program was run interactively,
# no output file is produced.
Output_File=c:\tools\sd\results1.txt

#*** OUTPUT FILE APPEND MODE ***
# The output file is normally in overwrite mode. To write in
# append mode, use the Append to Output File=1 setting.
#  0 = Overwrite
#  1 = Append mode
Append to Output File=0

#*** SUBDIRECTORY SEARCHING ***
# The choices for Search_subdir are
#  0 = Off
#  1 = On
# The default is 0.
Search_subdir=1

#*** DISPLAY ALL RESULTS OR JUST THOSE WITH PROBLEMS ***
# When "Show All" is enabled, the results for all files
# will be displayed. When it is disabled, only the results
# for shortcuts that have problems will be listed. The choices
# are:
#  0 = Show All is disabled
#  1 = Show All is enabled
Show All=0

#*** VERIFY PATHS BEFORE REPLACEMENTS ***
# The Check before replace= switch determine whether or not the
# program verifies paths in the Target, Start In, and Icon paths
# before allowing a replace in any of those fields. When enabled,
# if any of one cannot be verified, no replace is made for those
# fields. When Check before replace=0, no verification is attempted.
# The choices for the "Check before replace" switch are
#  0 = Do not check before replace
#  1 = Check before replace
# The default is 1 if this switch is not specified.
Check before replace=1

# *** DISPLAY PROGRESS METER ***
# To enable/disable the progress meter, use these settings:
#  0 = No progress meter is displayed.
#  1 = Show progress meter
# If not specified, Default = 0
Show Progress=0

# *** REPLACEMENT PROMPTING ****
# To enable/disable replacement prompts:
#  0 = No prompting
#  1 = Yes, prompting
Prompt On Replace=1

#*** CHANGE AUTOMATIC NETWORK SHORTCUT RESOLUTION ***
# By default, the program does not change the tracking setting
# for a network shortcut. You can enable/disable automatic resolution
# with the switches:
# Make Tracking=1
# or
# Make Non Tracking=1
# (Remove appropriate comment character above)

# *** ADDITIONAL ACTIONS ***
[Action2]
Path=c:\winnt
Search_string=d:\program files\tools\sr
Replace_string=f:\tools\sr
Output_File=c:\tools\sd\results2.txt
Search_subdir=1
Show_Files=0
Check before replace=1

[Action3]
Path=c:\winnt
Search_string=d:\tools\dirtkt
Replace_string=f:\tools\dirtkt
Search_subdir=1
```

Shortcut Operations - Automatic Network Shortcut Resolution

By default, Windows 95/98/NT attempts to automatically resolve the UNC path of shortcuts to mapped networked drives based on the original UNC network resource name at the time the shortcut was created. For example, if you create a shortcut to someprogram.exe on UNC path \\server1\c\program files mapped as drive M:, if at a later point \\server1\c\ is mapped as drive O:, Windows will automatically resolve the new network mapping when you use that shortcut.

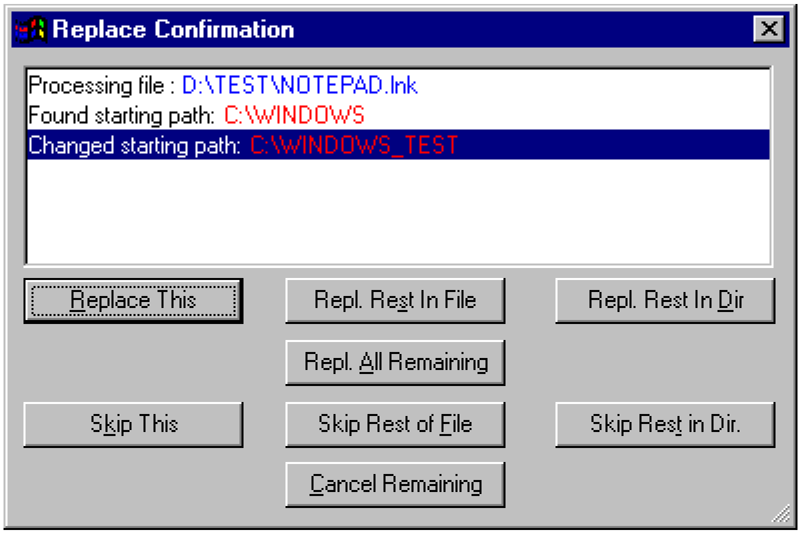
There may be situations where this behavior is undesirable. For example, if someprogram.exe expects mappings to drive M:, problems may occur.

Update Toolkit can enable/disable network tracking of a shortcut via the Make Tracking=1 or Make Non Tracking=1 switches in a Shortcut Operations script.

For further information on automatic network shortcut resolution, search the Microsoft Knowledge Base (support.microsoft.com). Article "Q150215" is a good starting point.

Shortcut Operations - Confirmation Prompt Dialog

When confirmation prompts are enabled for a Shortcut Operation via the `Prompt On Replace=1` switch, the below dialog is presented:



Processing file : D:\TEST\NOTEPAD.lnk
Found starting path: C:\WINDOWS
Changed starting path: C:\WINDOWS_TEST

Registry Operations Syntax

Listed below are the Update Toolkit script switches for registry operations.

```
# General Comments

# The # character in column 1 is the comment character.
# These must go in column 1.

# All switches must start in column 1.

# Scripts are divided into [Action ] sections that are
# numbered sequentially, e.g, [Action01], [Action02],
# [Action03], ....  They can be in any order in the
# script but are processed sequentially by number.

# Scripts can also have an optional [Default] section.
# [Default] can be used to specify switches that are to
# be used in all [Action ] that do not have that same switch
# specified.  A switch specified in an [Action ] always
# overrides the counterpart (if any) from [Default].

# All scripts MUST have at least one [Action ] section.
# For a single action script, DO NOT USE a [Default]
# section.  Use [Action1] instead.

# If your script has multiple actions and if you are
# using output file(s), use a different Output/Undo File=
# specification for each [Action ].  File writes to the
# output file are 'overwrite', not 'append'.

# See Export via a script for more information on exports.

[Default]

# Use this optional section to specify any switches to be
# applied to all [Action ]'s.  We suggest new users
# omit [Default] entirely and specify all switches within
# each [Action ].  You can streamline later if you want by
# making a [Default].

# *** SEARCH AND REPLCE STRINGS ***
# It is OK to leave Replace String= blank when conducting
# search-only operations.  Results of the search will be
# written to the undo file.
# If you wish to use the undo file to preview the effects
# of a replacement, enter a string in Replace String= and
# specify Replace=0 (see below).
# If Search String= and Replace String= are supplied in
# [Default], create an [Action1] section with no
# Search String= and Replace String= setting in order to use
# Search String= and Replace String= as defined in [Default]
# If your [Action1] includes a specific Search String= and
# Replace String= settings they will supercede what is defined
# for default.
Search String=Funduc
Replace String=

# *** HIVE/HEY TO SEARCH ***
# Specify a top level key name to begin the operation from.
Key=HKEY_CURRENT_USER\Software

# *** CASE SENSITIVE OPERATION ***
# 0 = Case Insensitive search
# 1 = Case Sensitive search
# If not specified, Default = 0
Case Sensitive=0

# *** WHOLE WORD OPERATION ***
# 0 = Whole Word matching for the search string off
# 1 = Whole Word matching search string on
# If not specified, Default = 0
Whole Word=0

# *** REGULAR EXPRESSION OPERATION ***
# See Registry Operations - Regular Expressions for more info.
# 0 = Regular Expression mode off
# 1 = Regular Expression mode on
# If not specified, Default = 0
Regular Expression=0

# *** OPERATIONS ON VALUES ***
# 0 = Do not search (or replace) in registry values
# 1 = Search (or replace) in registry values
# If not specified, Default = 0
Values=1

# *** OPERATIONS ON KEYS ***
# 0 = Do not search (or replace) registry keys
# 1 = Search (or replace) registry keys
# If not specified, Default = 0
Keys=1

# *** OPERATIONS ON DATA ***
# 0 = Do not search (or replace) data values
# 1 = Search (or replace) data values
# If not specified, Default = 0
Data=1

# *** OPERATIONS ON STRING DATA ***
# 0 = Do not search (or replace) REG_SZ data types
# 1 = Search (or replace) REG_SZ data types
# If not specified, Default = 0
String=1

# *** OPERATIONS ON EXPANDED STRING DATA ***
# 0 = Do not search (or replace) REG_EXPAND_SZ data types
# 1 = Search (or replace) REG_EXPAND_SZ data types
# If not specified, Default = 0
Expanded String=0

# *** OPERATIONS ON MUTLI LINE STRING DATA ***
# 0 = Do not search (or replace) REG_MULTI_SZ data types
# 1 = Search (or replace) REG_MULTI_SZ data types
# If not specified, Default = 0
Multi String=0

# *** OPERATIONS BINARY DATA ***
# 0 = Do not search REG_BINARY data types
# 1 = Search REG_BINARY data types
# If not specified, Default = 0
Binary=0

# *** PROCESS SUBKEYS OR NOT ***
# By default the program will recurse all subkeys. To
# process values/data only in the immediate key specified
# in the Key= switch above, use the Process Subkeys=0
# switch.
# 0 = Do not process subkeys
# 1 = Process Subkeys
Process Subkeys=1

# *** OUTPUT/UNDO FILE SPECIFICATION ***
# Note: If nothing is specified after the = character or you
# omit this switch completely, the program will create one or
# more undo/output files in the last path for those when the
# program was run interactively.
# If a specific file is indicted as modeled above, output is
# directed to that file.
# You can specify a different output/undo file for each [ActionX]
# section in a multi-step script.
# The Output/Undo File is NOT an append mode write.  You may want
# to specify a different Output/Undo File for each step so contents
# are not overwritten by a later step.
Output/Undo File=F:\FUNDUC\REGTKT\RTREG1.REG

# *** REPLACEMENT PROMPTING ***
# Note: When confirm is on, you may chose to skip all further
# confirmations in the confirmation dialog after the first prompt
# is offered.
# 0 = No initial replacement confirmation
# 1 = Confirm replacement prompt supplied for first replacement
# If not specified, Default = 0
Prompts=1

# *** PERMIT REPLACEMENTS? ***
# Use this when you have a Replace String specified but just want
# to preview the results of your replace. The preview will be
# listed in the Output/Undo File.
# 0 = No replace
# 1 = Perform a replace
# If not specified, Default = 0
Replace=0

# *** DISPLAY PROGRESS METER ***
# 0 = Standard progress meter
# 1 = No progress meter is displayed
# If not specified, Default = 0
No Progress=0

# *** WRITE REPLACEMENT STRING TO VALUE DATA ***
# To write the replace string where a value is found
# by the search string instead of changing the value
# itself, specify the data type with this switch.
# 1 = REG_SZ
# 2 = REG_EXPAND_SZ
# 3 = REG_BINARY
# 4 = REG_DWORD
# 4 = REG_DWORD_LITTLE_ENDIAN
# 5 = REG_DWORD_BIG_ENDIAN
# 6 = REG_LINK
# 7 = REG_MULTI_SZ
# If not specified, Default=0.
Set Value Data=1


# *** ADDITIONAL ACTIONS ***
# NOTE: The [ActionX] sections can be specified in any
# any order in the script file.  The program will parse
# the script file and implement them sequentially based
# on their number,
# e.g., [Action1], [Action2], [Action3], etc.

[Action1]
Key=HKEY_CURRENT_USER\Software2

[Action2]
Search String=Test
Replace String=Testing
Key=HKEY_CURRENT_USER\Software\Test
Action Replace=1
Case Sensitive=1
Whole Word=1
Regular Expression=0
Values=1
Keys=0
Data=0
String=1
Expanded String=0
Multi String=0
Binary=0
Output/Undo File=F:\FUNDUC\REGTKT\RTREG2.REG
```

```
Prompts=0
```

Registry Operations Syntax - Export Function

Registry data can be export to a text file (Regedit 4 style) via a Registry Operation using the format modeled below. When using scripts for an export, omit entirely the Search String= and Replace String= switches that are part of the typical script. Use the Key= switch to specify the top level to begin exporting from. All keys/data under that point are exported. Use the combination of value/data types switches modeled below. Be sure to specify an Output/Undo File. Also, be sure to specify Replace=0!

These special action export scripts should not be combined with other actions -- make a single export per script.

```
[Action1]
Key=HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall
Values=0
Keys=1
Data=1
String=1
Expanded String=1
Multi String=1
Binary=1
Output/Undo File=c:\temp\REGDUMP.REG
Replace=0
```

Registry Operations - Output/Undo Files

The output files created by a Registry Operation are the standard ' REGEDIT4' ascii .reg fies. You can direct the results of an operation to specific .reg file via the 'Output/Undo File=' switch. Alternatively, if you omit the 'Output/Undo File=' switch or leave it unspecified, the program will write multiple undo .reg file(s) in the past path used by the program when it was run interactively.

Registry Operations - Regular Expression Syntax

Registry Operations can use a regular expressions for advanced searches, where the search string follows a rule and is not always exactly the same. They employ "special character operators," which are symbols that control the search, and "an expression," which is a combination of characters and operators that specifies a pattern. Update Toolkit uses a Regular Expression syntax that is a subset of UNIX grep notation for searching and DOS style Batch file %n input parameter notation for replacing. The major differences between the Registry Toolkit and grep regular expressions are:

- **.** operator is not supported
- *** + ? !** operators must **precede** an expression enclosed by () or [].

Literal Characters:

- The following must be preceded by a \ character to be interpreted as a literal in a search:
  `- + * ? ( ) [ ] \ | $ ^ !`
- The following must be preceded by a \ to be interpreted as a literal in a replace :
  `% \ < >`

Do not forget to use `Regular Expression=1` in your script.

Regular expressions in registry scripts may be combined with Environment Variables.

Regular Expression Search - Match Operators

***** **Zero or More Operator:** Matches zero or more expressions enclosed in () or []. * may be used by itself, although it is intended to be used around strings. If the * operator is entered alone it will match all characters from the start of the line to the end of the line. Entering multiple expressions containing * should be done carefully to avoid overlapping matches & unpredictable results.

***(is)**	matches	**is**, **Mississippi**
***[is]**	matches	**Some**, **Expression**, **single**
Windows *[0-9]	matches	**Windows 95, Windows 98**

+ **One Or More Operator:** Matches one or more of the occurrences of the expression. + is intended to be combined with () and care should be taken when using + by itself. For example,

+(is)	matches	**is**, **Mississippi**
w+e	matches	**wide**, **white**, **write** but not **we**

? **One Occurrence Operator:** Matches exactly any character either before or after a string. ? also matches any character between two strings. When combined with (), ? matches exactly one expression enclosed in (). Using the ? operator by itself will match every character and should be avoided.

?(is)	matches	**is** or **si**
Win?95	matches	**Win 95**, **Win-95**, **Win/95**

| **Or Operator:** Matches the simple expression either before or after the | (pipe) symbol. This should be used in conjunction with (). Or expressions should not contain other operators such as *+^$?. You may, however, make use of other operators outside the (). For example,

<table>
<tr><td><b>(01/|02/)+[0-9](/95|/98)</b></td><td>matches</td><td><b>01/15/98</b> &amp; <b>02/12/98</b> but <b>not 03/12/98</b></td></tr>
<tr><td><b>w*[a-z](98|NT)\?</b></td><td>matches</td><td><b>Windows 98?</b>, <b>Win NT?</b></td></tr>
</table>

! **Not Operator:** No match will be made if the expression is found. The NOT operator can be combined with other wildcard & subexpression operators. When combined with [], nothing in the range described in the brackets will be matched. For example,

!(b|c)at	matches	**at** in **mat**, **rat** or **fat** but not in **bat** or **cat**
!(Windows)98	matches	**98** in **1998** but not in **Windows 98**

^ **Beginning Of Line Operator:** Matches an expression that starts at the beginning of a line. This may be combined with other wildcard and subexpression operators. However, avoid using ^ inside () expressions.

^the	matches	**the**, **The**, **THE**, **tHE** at the beginning of a line

$ **End Of Line Operator:** Matches an expression that is found at the end of a line. This may be combined with other wildcard and subexpression operators. However, avoid using $ inside () expressions.

end$	matches	**end** only if it is at the end of a line

Regular Expression Search - Sub-Expression Operators

[ ] **Range Operator:** This may be a list of single characters such as [dog], one or more ranges of characters such as [d-o0-2], or a more complex expression using other match or sub-expression operators such as do[g|uble]. Ranges are allowed by using the "a-z" notation. When entered alone, [] will match all characters and is equivalent to ?[]. Use the ?*+ operators to modify the range to be matched by the [] sub-expression operator. You can also enter [a-zA-Z0-9\-_], e.g., multiple ranges and the - character itself preceded by a \. For example,

t[]e	matches	**The**, **Toe**
H*[]d	matches	**Hello** (cr -lf) **World** across two lines
***[0-9]**	matches	**234907**, **5795**, or an empty string
+[niewW]	matches	one or more strings such as **Win**, **new**, **win**
[a-z]	matches	any lower case string if case sensitive is on and any words if not case sensitive

( ) **Subexpression Operator:** Parentheses are used to denote one or more sub-expressions. This is usually combined with the OR operator - \|. For example,

Win(95|dows 95)	matches	**Windows 95** and **Win 95**

Regular Expression Replacement Operators

%n **Basic Replacement Operator:** Core replacement operators use a %n convention, where n corresponds to a component in the regular expression search string. For example, %1 refers to the first expression value in the search string, %2 refers to the second, etc. %n parameters may be used several times, omitted, or used in any order. Up to 24 parameters may be used at once by referring to those over number %9 moving up the ASCII table, e.g., 123456789:;<=>?@ABCDEFGH. Some examples are:

Given:	http://www.funduc.com http://www.searchandreplace.com
Search Expression:	http://*
Replacement Expression:	%1
Results:	www.funduc.com www.searchandreplace.com

Given:	APPL 33.175 C 65.937
Search Expression:	*[a-z] *[0-9].*[0-9]
Replacement Expression:	%1,%2.%2
Results:	APPL,33.175 C,65.937

<   **Lower Case Operator:** This is to be used in conjunction with %n, where %n< replaces the original matched expression with the lower case version. For example,

Given:	MAKE THIS ALL LOWER CASE
Search Expression:	+[A-Z]
Replacement Expression:	%1<
Results:	make this all lower case

>   **Upper Case Operator:** This is to be used in conjunction with %n: For example, %1> replaces the original first matched expression with the upper case version. For example,

Given the series:	All words beginning with letter w.
Search Expression:	w+[a-z]
Replacement Expression:	W%1>
Results:	All WORDS beginning WITH letter w.

Registry Operations - Environment Variables

Registry Operations can make use of environment variables in the `Search String=` and `Replace String=` switches in a script. Searches may be case sensitive. Environment variables can also be combined with regular expressions.

The syntax is:

%%envvar=variable name%%

where variable name is the name of the environment variable to use.

For example, to search for the value of the environment variable *temp*, enter the string

%%envvar=temp%%

in the search field or your regular expression. If the value of your *temp* environment variable is *c:\windows\temp*, search hits would occur wherever the string *c:\windows\temp* is found in the registry. If Case Sensitive is on, *c:\windows\temp* would be found but *C:\WINDOWS\TEMP* would not.

To insert the value of the environment variable *winbootdir* in a replacement string, enter the string

%%envvar=winbootdir%%

in the replace string field or your regular expression. If the value of your *winbootdir* variable is *C:\WINDOWS*, the string *C:\WINDOWS* would be used in replacements.

A regular expression operation involving the environment variables operators might be:

Search:	*%%envvar=tmp%%
Replace	%1TMP=%%envvar=tmp%%;%%envvar=winbootdir%%

Text File Search and Replace Syntax

Listed below are the script switches for text file search and replace operations.

Please note that the program ALWAYS replaces. You can enable confirmations via the `Prompt_flag=0` or `Prompt_flag=1` switch to disallow replaces. You cannot use Update Toolkit to list search results-only. Our Search and Replace utility should be used if you need to do general searching.

Update Toolkit writes a file named 'SRUNDO.bat' in the same path as updtkt.exe if you have a backup path defined using the `Backup Path=` switch (see below). SRUNDO.bat can be used to reverse the last S-R made.

Search-Replace on ZIP file contents is not supported.

We assume that Update Toolkit users are advanced computer users who are well aware of the hazards of making replaces in proprietary and/or binary files such as Word .doc's, Spreadsheets, pdf files, Access databases, and so on. If you make replaces in these types of files do so at your own risk. It is generally the rule that replaces are safe provided they the replace string is exactly the same length as the search string. Nevertheless, experiment carefully on non-critical data before putting your operation into production use.

See Iteration Operator for information on the Script Loop= switch.. See Expression Evaluator for information in the [Expression] section. The [Options] section is optional. See Search Replace Options for additional information on the [Options] section switches. See Regular Expression Search Operators and Regular Expression Replace Operators for information on regular expression syntax.

Also see Special Replacement Operators for some special variables that can be specified in replacement strings. For example, you can output file dates, the entire search string, paths, and more.

Please note that if you specify linked script(s), follow-up scripts operate in the same mode as the script that launches them.

```
[Script for Search and Replace]
All text file search-replace scripts begin with the above tag.

You can insert comments in two places:
  1. The area here where you see this text, immediately after the
     [Script for Search and Replace] tag. Comments in this area must
     precede any other tags.
  2. On the same line as the [Search ] and [Replace] tags. See below
     for model.

There is no general comment character for text Search Replace scripts.

There should be no stray blank lines between the switches anywhere in
your script. Blank lines in the [Options] section will cause problems.
You can have multiple lines in the [Search] and [Replace] sections. If
you do you should specify '/b', for 'Binary'.

The following switches can be added to the [Search] tag:

  [Search /i]   Case Insensitive Search
  [Search /w]   Whole Word Search
  [Search /h]   Ignore Whitepace Mode
  [Search /b]   Binary Mode
  [Search /x]   Regular Expression Mode
Regular Expression can only be combined with /i. Other switches can be
combined with each other.

The following is the only switch for the [Replace] tag:

  [Replace /b]   Binary Mode

[Expression]
E1 AND E2
[Options]
Search_subdir=0
Prompt_flag=0
Output_File=
Show_Files=1
Backup Path=E:\TEMP\SRBACK\REPL
```

```
Before Hit=<
After Hit=>
Max Reg Expr=32767
Write to Backup Dir=0
Show Files Without Hits=1
Display Replace String=1
Display File Stats=1
One hit=0
Reverse Filters=0
Min Size Filter=0
Max Size Filter=0
Min Date Filter=0
Max Date Filter=0
Skip Files Mask=
Ignore Attributes=
Keep file time stamp=0
Sort File Names=3
Sort Ascending=0
Append to output file=0
Script Loop=1
[Search]          This comment added to a search line.
Search string #1
[Replace /b]      This comment added to a replace line.
Replacement string #1
[Search /bix]
Search string #2
[Replace]
Replacement string #2
[Path]
c:\*.ini
[Path]
d:\work\*.txt
[End of Search and Replace Script]
c:\tools\sr\Linked Script1.srs
c:\tools\sr\Linked Script2.srs
```

```
// Max Display Chars=512
Output_File=D:\FUNDUC\UPDTKT\Samples\outSR1.txt
// String Color=255
// Show_Files=1
// Max Items Saved=8
// File Color=16711680
Backup Path=D:\TEMP\SRBACK\
// Num Detail Entries=20
// Default Viewer=C:\Program Files\TextPad 4\TextPad.exe %path%(%line%,1)
// Before Hit=<
// After Hit=>
// Max Reg Expr=32767
Write to Backup Dir=0
// Unzip Dir=C:\WINDOWS\TEMP\
// Show Files Without Hits=0
// Display Replace String=0
```

Text File Search and Replace Syntax - Options Section

Below is some detail on the items in the [Expression] and [Options] section of a text Search Replace script. Use this as a guide for manual edits.

It is not necessary to specify all switches in the [Options] section. For example, a sole [Options] item could be Prompt_flag=0.

The order of the [Options] switches should be maintained to ensure the program parses the input properly. For example, if you have Show_Files= and Display Replace String= specified in your script, do so in this order.

Script Switch	**Notes**
[Expression]	
E1 AND E2	This section is optional. See Expression Evaluator for more information.
[Options]	
Search_subdir=	Determines whether or not to search in subdirectories. 0 = No; 1 = Yes. The default = 0.
Prompt_flag=	Determines whether you will be prompted for replacements. 0 = Prompt on All 1 = Prompt on Each File 2 = No Prompts.
Output_File=	Determines the complete path to an output file if you want. Omit or leave blank if none is to be used.
Show_Files=	Determines whether the search hits will be listed in the output file. The values are: 0 = No; 1 = Yes.
Backup Path=	Determines the path to use for file backups during a replace. The last change made to a file will be written to this path.
Before Hit= After Hit=	The delimiter characters to use before and after search hits in the Output File. This has an effect only when Show_Files=1.
Max Reg Expr=	Determines the maximum number bytes to span for a regular expression. If not specified the program default is 32 K.
Write to Backup Dir=	Determines whether or not the program writes replaced files to the Backup Path or in the original file path. When enabled, the files in the original path are left unchanged and the new files are written to the Backup Path. The values are: 0 = No; 1 = Yes.
Unzip Dir=	Determines the path to use for searching zip files.
Show Files Without Hits=	Determines whether or not the names of files without search hits are listed in the Output file. The values are: 0 = No; 1 = Yes.
Display Replace String=	Determines whether or not the replacement string is listed in the Output file. The values are: 0 = No; 1 = Yes.
Display File Stats=	When enabled, a count of the number of search

hits in a file is included in the Output file. The values are: 0 = No; 1 = Yes.

One hit=

When 'One hit' is enabled, the program will stop processing a file at the first search hit. This is useful if you only wish to list files that have at least one hit. When enabled, a replace operation will make only that one change so you may want to disable during replaces.
The values are: 0 = No; 1 = Yes.

Reverse Filters=0
Min Size Filter=
Max Size Filter=
Min Date Filter=
Max Date Filter=
Skip Files Mask=
Ignore Attributes=

These optional switches can be used to exclude or include files by date, size, or attribute. The syntax can be very complex. See Search Replace Filters for more information.

Keep file time stamp=

When enabled, the current time/date stamp for a file is preserved during a replace. The values are: 0 = No; 1 = Yes.

Sort File Names=

Use this to control the sort order of files in the Output report. The values are:
0 = Unsorted
1 = File Name
2 = File Extension
3 = File Date/Time
4 = File Size
5 = File Path.

Sort Ascending=

Use this to sort files ascending or not.
0 = Descending, 1 = Ascending.

Append to output file=0

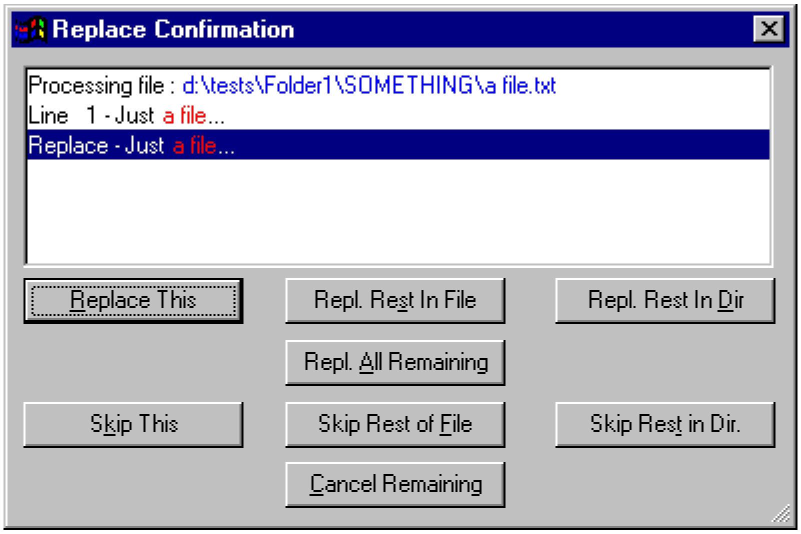
Enable this to write to the Output File in 'append mode'. The values are:
0 = No; 1 = Yes.

Script Loop=

This must be the last entry in the [Options] section. See Iteration Operator for more information.

Text File Search and Replace Syntax - Replace Confirmation Dialog

When the a text Search Replace uses the `Prompt_flag=0` or `Prompt_flag=1` switches the following dialog is presented:



```
Processing file : d:\tests\Folder1\SOMETHING\a file.txt
Line   1 - Just a file...
Replace - Just a file...
```

Script Item	**Notes**
[Expression] E1 AND E2	This section is optional. See Expression Evaluator for more information.
[Options] Search_subdir=	Determines whether or not to search in subdirectories. 0 = No; 1 = Yes. The default = 0.
Prompt_flag=	Determines whether you will be prompted for replacements. 0 = Prompt on All 1 = Prompt on Each File 2 = No Prompts.
Output_File=	Determines the complete path to an output file if you want. Omit or leave blank if none is to be used.
Show_Files=	Determines whether the search hits will be listed in the output file. The values are: 0 = No; 1 = Yes.
Backup Path=	Determines the path to use for file backups during a replace. The last change made to a file will be written to this path.
Before Hit= After Hit=	The delimiter character to use before and after search hits in the Output File. This has an effect only when Show_Files=1.
Max Reg Expr=	Determines the maximum number bytes to span for a regular expression. If not specified the program default is 32 K.
Write to Backup Dir=	Determines whether or not the program writes replaced files to the Backup Path or in the original file path. When enabled, the files in the original path are left unchanged and the new files are written to the Backup Path. The values are: 0 = No; 1 = Yes.
Unzip Dir=	Determines the path to use for searching zip files.
Show Files Without Hits=	Determines whether or not the names of files without search hits are listed in the Output file. The values are: 0 = No; 1 = Yes.
Display Replace String=	Determines whether or not the replacement string is listed in the Output file. The values are: 0 = No; 1 = Yes.
Display File Stats=	When enabled, a count of the number of search hits in a file is included in the Output file. The values are: 0 = No; 1 = Yes.
One hit=	When 'One hit' is enabled, the program will stop processing a file at the first search hit. This is useful if you only wish to list files that have at least one hit. When enabled, a replace operation will make only that one change so you may want to disable during replaces. The values are: 0 = No; 1 = Yes.
Reverse Filters=0 Min Size Filter= Max Size Filter= Min Date Filter= Max Date Filter= Skip Files Mask= Ignore Attributes=	These optional switches can be used to exclude or include files by date, size, or attribute. The syntax can be very complex. See Search Replace Filters for more information
Keep file time stamp=	When enabled, the current time/date stamp for a file is preserved during a replace. The values are: 0 = No; 1 = Yes.
Sort File Names=	Use this to control the sort order of files in the Output report. The values are: 0 = Unsorted 1 = File Name

	2 = File Extension 3 = File Date/Time 4 = File Size 5 = File Path.
Sort Ascending=	Use this to sort files ascending or not. 0 = Descending, 1 = Ascending.
Append to output file=0	Enable this to write to the Output File in 'append mode'. The values are: 0 = No; 1 = Yes.
Script Loop=	This must be the last entry in the [Options] section. See Iteration Operator for more information.

Text File Search Replace - Filters

Here are some details and example settings on the Max Size Filter=, Min Date Filter=, Max Date Filter=, Filter Skip Mask=, Ignored Attributes Mask= and Reverse Filters= switches for text Search Replace scripts.

```
#       FILTER SETTINGS FOR TEXT FILE SEARCH - REPLACE

NOTE: This material is NOT a sample script. DO NOT submit this to
the program as a script.

The number of combinations of values is extensive for Skip Files
Mask=, Ignore Attributes=, and Reverse Filters=. Some example
settings are below. Additional settings are possible. The Folder
Synchronize Script Editor can be used to generate a script. You
can also experiment with the Directory Toolkit Options-Filter
dialog and establish a file list output that meets your needs.
Then use the Directory Toolkit 'Write Options To Script' function
(View Menu) to export your settings to a script. Use your text
editor to get the Max Size Filter=, Min Date Filter=, Max Date
Filter=, Skip Files Mask=, Ignore Attributes= and Reverse Filters=
settings from that exported script into your production script.

It is also OK to write us at support@funduc.com ;-)

# *** SIZE & DATE EXCLUDE FILTERS ***
# Size & date filters exclude files by size &/or date. Any files
# that are less than the minimum size (in bytes) will be excluded
# from the operation. If you use Reverse Filters=1, the settings
# work as 'include' filters. Dates use the format specified in
# Control Panel (Regional Settings). Leave blank or omit entirely if
# you want no setting.
# Min Size Filter=0
# Max Size Filter=100000
# Min Date Filter=8/21/1998
# Max Date Filter=8/23/1998

# *** REVERSE FILTERS ***
# Reverse filters= is used to make the other filter switches
# function as exclude or include filters. For example, Reverse
# Filters=0 & Max Date Filter=01/01/2000 excludes all files with a
# stamp after Jan 01, 2000. Reverse Filters=1 changes this to
# include only files dated Jan 01, 2000 or after.
# The values are:
# 0 = Do not use Reverse Filtering.
# 1 = Reverse Filter mode enabled.
# If not specified, Default = 0.
Reverse Filters=0

# *** FILE & DIRECTORY ATTRIBUTE FILTERS ****
#
# The Skip Files Mask= and Ignore Attributes= switches are used
# together, in conjunction with Reverse Filters= to exclude/include
# some file types in the operation.

# The default values of Skip Files Mask=0 & Ignore Attributes=55
# mean no files or subdirs are filtered. If you wish to use these
# defaults it is not necessary to specify these switches.

# Please use the below list for some common settings.

# *** SOME SAMPLE SETTINGS ****

# Paste all three script switches into your script.

# Include files that are Read-Only along with any other attribute
Skip Files Mask=1
Ignore Attributes=54
Reverse Filters=1
```

```
# Exclude files that are Read-Only along with any other attribute
Skip Files Mask=1
Ignore Attributes=54
Reverse Filters=0

# Include files that are solely Read-Only
Skip Files Mask=1
Ignore Attributes=16
Reverse Filters=1

# Exclude files that are solely Read-Only
Skip Files Mask=1
Ignore Attributes=16
Reverse Filters=0

# Include Archive along with any other attribute
Skip Files Mask=32
Ignore Attributes=23
Reverse Filters=1

# Exclude Archive along with any other attribute
Skip Files Mask=32
Ignore Attributes=23
Reverse Filters=0

# Include solely Archive
Skip Files Mask=32
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Archive
Skip Files Mask=32
Ignore Attributes=16
Reverse Filters=0

# Include System along with any other attribute
Skip Files Mask=4
Ignore Attributes=51
Reverse Filters=1

# Exclude System along with any other attribute
Skip Files Mask=4
Ignore Attributes=51
Reverse Filters=0

# Include solely System
Skip Files Mask=4
Ignore Attributes=16
Reverse Filters=1

# Exclude solely System
Skip Files Mask=4
Ignore Attributes=16
Reverse Filters=0

# Include Hidden along with any other attribute
Skip Files Mask=2
Ignore Attributes=53
Reverse Filters=1

# Exclude Hidden along with any other attribute
Skip Files Mask=2
Ignore Attributes=53
Reverse Filters=0

# Include solely Hidden
Skip Files Mask=2
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Hidden
```

```
Skip Files Mask=2
Ignore Attributes=16
Reverse Filters=0

# Include Read-Only & Hidden along with any other attribute
Skip Files Mask=3
Ignore Attributes=52
Reverse Filters=1

# Exclude Read-Only & Hidden along with any other attribute
Skip Files Mask=3
Ignore Attributes=52
Reverse Filters=0

# Include solely Read-Only & Hidden Alone
Skip Files Mask=3
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Read-Only & Hidden
Skip Files Mask=3
Ignore Attributes=16
Reverse Filters=0

# Include Read-Only, Hidden, & System along with any other attribute
Skip Files Mask=7
Ignore Attributes=48
Reverse Filters=1

# Exclude Read-Only, Hidden, & System along with any other attribute
Skip Files Mask=7
Ignore Attributes=48
Reverse Filters=0

# Include solely Read-Only, Hidden, & System
Skip Files Mask=7
Ignore Attributes=16
Reverse Filters=1

# Exclude solely Read-Only, Hidden, & System
Skip Files Mask=7
Ignore Attributes=16
Reverse Filters=0

# Include files with No Attribute Setting
Skip Files Mask=0
Ignore Attributes=16
Reverse Filters=1

# Exclude files with No Attribute Setting
Skip Files Mask=0
Ignore Attributes=16
Reverse Filters=0
```

Text File Search Replace - Expression Evaluator

The "Expression Evaluator" section in a text Search Replace script is an optional setting that can be used to determine which files among those meeting the path and filter criteria are processed by your script. The setting, if used, is on the line immediately after the `[Expression]` tag in a script.

The expression uses the variables, `E1, E2, ... En`, to reference the search/replace string pairs in the script. A script with 3 strings could use `E1, E2, & E3` available.

The `Ex` variables are separated by one of the following operators:

```
AND OR NOT < > <= >= >+ = != + - * /
```

By way of example, the default expression assumed by the program if `[Expression]` is an OR operation. If your script has 3 search/replace pairs, the program uses

```
[Expression]
E1 OR E2 OR E3
```

In words, this means that if the program files any of the search strings in a file, that file is processed.

Numeric values may be used in conjunction with variables to denote the number of replacement opportunities in a file. For example, the expression

```
[Expression]
E1 AND (E2 = 4)
```

would process only those files where the first search & replace pair could occur one (or more) times and the second search & replace pair could occur exactly 4 times.

Notes:

- The `[Expression]` section is intended for use with replacement prompting turned off. Be sure to read the help section on Expression Evaluator and Replacement Operations for information about the behavior of replacement prompts when a complex expression is in effect and advice on debugging your expressions.
- Unary versions of the + and - operators must be enclosed in parentheses, e.g., (-1).
- You can compare one string to another, e.g.,
  ```
  E1 AND NOT(E2 != (E3)))
  E1 AND NOT(E2=1 or (E3 != (-1)))
  ```
- You can use arithmetic operations involving En values, e.g.,
  ```
  (E1 * 2) = E3
  ```
- `E1` and (`E1 >= 1)` are equivalent. Conversely, `NOT E1` and (`E1 = 0)` are equivalent.
- Operators are parsed in the following order:
  ```
  1) (
  2) * /
  3) + -
  4) = != > >= < <=
  5) NOT
  6) AND
  7) OR
  ```

Some Basic Expression Evaluator Examples:

(Note: These examples assume the user either: a) has Replace Prompts turned off or b) does not "Skip" any replacements if Replace Prompts are turned on. See Expression Evaluator and Replacement Operations for more information about the possible influence of "Skip(s)" on expressions involving numeric values.)

FILE CONTENTS		
File 1	**File 2**	**File 3**

JILL	JACK	JILL
JACK	CAROL	JACK
JOHN		JOHN

Script Search Strings:	E1 = JILL
	E2 = JACK
	E3 = JOHN
	E4 = CAROL

Expression Evaluator:	E2 AND E4
Files That Would Be Processed:	FILE 2

FILE CONTENTS

File 1	**File 2**	**File 3**
JILL	JACK	JILL
JACK	CAROL	JACK
JOHN		JOHN

Script Search Strings:	E1 = JILL
	E2 = JACK
	E3 = JOHN
	E4 = CAROL

Expression Evaluator:	E1 AND E4
Files That Would Be Processed:	(none)

FILE CONTENTS

File 1	**File 2**	**File 3**
12/18/94	12/01/95	12/19/95
Page 95	12/15/95	www.funduc.com
www.funduc.com		

Script Search Strings:	E1 = 12
	E2 = /*[0-9]/
	E3 = 95
	E4 = www.funduc.com

Expression Evaluator:	E1 AND E2 AND ((E3 >= 2 OR E4)
Files That Would Be Processed:	FILE2 FILE3

Text File Search and Replace - Expression Evaluator & Replacement Prompts

Expression Evaluator is geared toward complex operators that are run with replacement prompts turned off. It is OK to use Replacement Prompting but please note that the behavior of replacement prompts during search & replace operations differs from that of an operator where `[Expression]` is not used.

Specifically, the Expression Evaluator is applied after all the search hits and 'replacement opportunities' for the entire operation are accumulated.

When replacement prompts are on (Prompt_flag=0), prompts are offered for search hits for any of the search strings, even though only some of those files might satisfy the criteria in the `[Expression]` criteria. This can be unnerving because seems like the program will make a replace in a file that does not meet the criteria. As long as you confirm the replacement prompt, if the file does not meet the `[Expression]` criteria, when it comes time to make the actual replace, replacements will happen only in files that satisfy the `[Expression]` criteria.

In fact, it is important that you permit all replacement prompts. If you respond with "Skip" to a replacement that initially satisfies your expression, the "Skip" may influence the result of your expression. Number values in expressions are based on the number of search & replace opportunities in a file. Since a "Skip" eliminates one replacement opportunity, "Skip(s)" may impact expressions that make use of number values. For example, given the expression (`E1 >= 2`) and a file having exactly 2 search & replace opportunities. A manual "Skip" in response to either of the replacement prompts eliminates 1 search & replace opportunity. The file would therefore no longer satisfy the expression (`E1 >= 2`). This in turn means that no replacements would be made, even if you respond "Replace This" to the other prompt.

In some cases you may wish to enable replacement prompts in order to capitalize on the above behavior as this provides a way to have 'on the fly' control over your expressions.

Regarding debugging expressions, you should note that search-only operations report hits only in files that actually satisfy the expression. Therefore, to debug your script in advance of a search & replace operation and identify the specific files that will be selected by your expression, run a search-only operation. Another debugging tactic is to intentionally run a search & replace operation with confirmations set to "Prompt on all". This allows you to see all search/replace strings in-context in all files.

The table below outlines these characteristics:

REPLACEMENT PROMPTS & ADVANCED SCRIPT SETTINGS

Given the following:

Contents of File #1:	12/18/94 Page 12
Contents of File #2:	12/19/95 www.funduc.com

Script Search Strings:	E1 = 12
	E2 = Funduc

Expression Evaluator:	E1 AND E2

Results of the following scenarios would be:

Operation	**Results**
Search-Only:	The strings "12" and "funduc" would be reported in File #2 only.
Unprompted Search & Replace:	Replacements would be made in File #2 only.
Prompted Search & Replace: (Where user selects "Replace This" in each instance)	You would be prompted to replace the "12" in File #1 **and** "12" & "funduc" in File #2. Even if you answer "Replace This" to the "12" in File #1, File #1 would not be modified because it does not satisfy the Expression, (E1 AND E2). File #2 would be modified because you allowed all replacements to occur in File #2.
Prompted Search & Replace: (Where user selects "Skip" to either of the prompts for File #2)	No replacements would occur in File #1 (as above). No replacements would occur in File #2 because a skip in response to either prompt for File #2 means it no

longer satisfies the expression (E1 AND E2).

Text File Search and Replace - Iteration Operator

The `Script Loop=` switch in a text Search Replace script is an optional setting that is used when a search/replace string might not make all possible replacements on the first pass. This can happen, for example, when a * or *[] search operator in your overall regular expression search term spans text that includes a string component that you wish to have replacements done on. This can also happen if a replacement somehow impacts the next instance of a string that would be found by the search term.

Non-Regular Expression search/replaces would likely not be affected in this way.

The value can be larger than necessary. If no more search hits are found the program will stop the iterations.

Usage:

Script Loop=x

> where x = the number of times to repeat the script. The program does not have an internal limit for the maximum value. Use a value of -1 to cause the script to repeat indefinitely (not recommended). Press the ESC key to abort a running script.

This switch **must be** the last item in the [Options] section of the script.

Example Script using Script Loop=

```
[Script for Search and Replace]
[Options]
Script Loop=5
[Search /ix]
^\r\n\r\n\r\n
[Replace]
\r\n\r\n
[End of Search and Replace Script]
```

Text File Search and Replace - Linked Scripts

Multiple text Search Replace scripts can be chained from a single script by specifying one or more script files at the bottom of a script, after the `[End of Search and Replace Script]` tag.

Please read through this entire section before you use Script Linking for the first time!

Example Script with Linked Scripts

```
[Script for Search and Replace]
[Search /i]
something
[Replace]
something
[Path]
D:\TEST\somefile.txt
[Path]
F:\TESTING\*.htm*
[End of Search and Replace Script]
D:\TEST\Script2.srs
D:\TEST\Script3.srs
D:\TEST\Script4.srs
...(Additional scripts)...
D:\TEST\ScriptLast.srs
```

Notes:

- Linked scripts perform the same 'search-only' vs. 'search/replace' operation as that in effect for script that launches them. For example, if the initial script containing link references to other script(s) is launched as a search-only operation, the chained scripts will also perform a search-only. Conversely, if the initial script is launched as an search-replace, the other scripts will also perform a search-replace. See below for usage suggestions.
- Avoid nested Linked Script references!! For example, if script1.srs chains to script2.srs, do not place any linked script references in script2.srs.
- Individually debug all scripts in advance. There is no way to 'pause' a series of linked scripts.

Text File Search and Replace - Regular Expression Search Operators

Listed below are the regular expression search operators available for text Search Replace scripts.

Notes:

- The following characters must be preceded by a \ character to be interpreted as a literal in a search using regular expressions: – + * ? ( ) [ ] \ | $ ^ !
- We assume that Update Toolkit users are advanced computer users who are well aware of the hazards of making replaces in proprietary and/or binary files such as Word .doc's, Spreadsheets, pdf files, Access databases, and so on. If you make replaces in these types of files do so at your own risk. Experiment carefully in advance.

Regular Expression Search - Match Operators

* **Zero or More Operator:** Matches zero or more expressions enclosed in () or []. * may be used by itself, although it is intended to be used around strings. If the * operator is entered alone it will match all characters from the start of the line to the end of the line. You can match characters between two or more strings up to the Max Reg Expr= setting by specifying a range after the * operator. Entering several expressions in a row containing * should be done carefully to avoid overlapping matches which may produce unpredictable results.

*(is)	matches	is, Mississippi
*[is]	matches	Some, Expression, single
Windows *[0-9]	matches	Windows 95, Windows 98
Windows*[]95	matches	up to 32767 characters (across several lines) between Windows and 95
Windows*[\0-ÿ]	matches	same as the above (older syntax)

Note: When * is combined with a numeric range and the *%n>>* or *%n>starting value>* replacement operators, the search expression above, Windows *[0-9], would be part of a Counter Operation.

+ **One Or More Operator:** Matches one or more of the occurrences of the expression. + is intended to be combined with () and care should be taken when using + by itself. For example,

+(is)	matches	is, Mississippi
w+e	matches	wide, white, write but not we

? **One Occurrence Operator:** Matches exactly any character either before or after a string. ? also matches any character between two strings. When combined with (), ? matches exactly one expression enclosed in (). Using the ? operator by itself will match every character in a file one at a time and therefore probably should be avoided.

?(is)	matches	is
Win?95	matches	Win 95, Win-95, Win/95

| **Or Operator:** Matches the simple expression either before or after the | (pipe) symbol. This should be used in conjunction with (). Or expressions should not contain other operators such as *+^$?. You may, however, make use of other operators outside the (). For example,

(01/|02/)+[0-9](/95|/98)	matches	01/15/98 & 02/12/98
w*[a-z](98|NT)\?	matches	Windows 98?, Win NT?

! **Not Operator:** A match will be made when both a 'positive' hit component and a !() or ![] component of the expression are found. The complete expression requires both components. The first may be as simple as a single regular expression operator such as * or ?. You should provide a wild card operator of some type prior to the ! component. The ! component should be enclosed in () or []. Be sure to nest ( ) when using |, e.g., ?!

(a|b) won't work - Use ?!((a|b)) instead. Multiple!() component can be used to create an 'or', e.g, ?!(a)!(b)!(c). You can also use other regular expressions inside ( ). Additional 'postive hit' strings &/or regular expressions to find may be specified after !() or ![]. Note, however, that regular expressions following the !() or ![] will not be available to the %n operators. See Not Operator Notes for more information.

?at!((b|c)at)	matches	mat & sat but not 'b'at or 'c'at
*file!(beg*file)	matches	a file & this file but not 'beginning of file'
*98!(Windows 98)	matches	98 in 1998 but not in 'Windows 98'
*98!(+[a-z]98)	matches	98 in 1998 but not in 'Windows 98'
a?b?c!(aub?c)!(a?bvc)	matches	In a search of 'aXbYc', find where X not 'u' and Y not 'v'
^*!(^t)!(</p>$)	matches	Lines that don't begin w/ t or end with </p>
#include*!()"C	matches	"ChildFrm.h" in "#include "ChildFrm.h"" but finds nothing in "#include "Edit.h"". This example uses !() as an AND operator of sorts.

^ **Beginning Of Line Operator:** Matches an expression that starts at the beginning of a line. ^ should be the first character in your expression. ^ can be be combined with other wildcard and subexpression operators. However, except for the 'not' operator, do not use ^ inside () expressions. Use *(\r)\n instead. Also, do not combine ^ and $ in a single expression.

Note: The ^, $, ^^, and $$ operators are counted for the purposes of an %n operator in a replacement expression. For example, in the search expression ^+[ ][a-zA-Z], the corresponding %n terms are: %1 = ^, %2= +[ ], %3 = [a-zA-Z].

Some examples of ^ are:

^the	matches	the, The, THE, tHE at the beginning of a line and if case sensitive is off.
^(the|[a<])	matches	the in The opening paragraph... A in At 5 pm ... < in <HR>
^the*end.\r\n	matches	And entire line that begins with The and ends with end.

$ **End Of Line Operator:** Matches an expression that is found at the end of a line. $ should be the last character in your expression. $ can be use with other wildcard and sub expression operators. However, except for the 'not' operator, do not use $ inside () expressions. Use *(\r)\n instead. Also, do not combine ^ and $ in a single expression.

Note: The ^, $, ^^, and $$ operators are counted for the purposes of an %n operator in a replacement expression. For example, in the search expression l+[ls]$, corresponding %n terms are: %1 = +[ls], %2= $.

Some examples of $ are:

end$	matches	end only if it is at the end of a line
(the end|of a line.)$	matches	the end in "This is the end" of a line. in "the end of a line. "

^^ **Beginning Of File Operator:** Matches an expression found at the beginning of a file. Usage is similar to ^. Do not use ^^ inside ().

Note: The ^, $, ^^, and $$ operators are counted for the purposes of an %n operator in a replacement expression. For example, in the search expression ^^?omething, the corresponding %n terms are: %1 = ^^, %2= ?.

^^First	matches	First in "First line of the file" if that string is on the first line of the file.
^^+50[]	matches	The first 50 characters in the file.

$$ **End Of File Operator:** Matches an expression found the end of the file. Usage is similar to $. Do not use $$ inside ()

Note: The ^, $, ^^, and $$ operators are counted for the purposes of an %n operator in a replacement expression. For example, in the search expression in the below, *$$, the corresponding %n terms are: %1 = *, %2= $$.

*$$	matches	The last line in the file

Regular Expression Search - Sub-Expression Operators

[ ] **Range Operator:** This may be a list of single characters such as [dog], one or more ranges of characters such as [d-o0-2], or a more complex expression using other match or sub-expression operators such as do[g|uble]. Ranges are allowed by using the "a-z" notation. When entered alone, [] will match all characters and is equivalent to ?[]. Care should be used with [] alone as it will match all characters in the file. When combined with the ***** operator, as in *[], the match will span across one or more lines up to the number of characters specified in the Max Reg Expr= setting. Use the ?*+ operators to modify the range to be matched by the [] sub-expression operator. You can also enter [a-zA-Z0-9\-_], e.g., multiple ranges and the - character itself preceded by a \. For example,

t[]e	matches	The, Toe
H*[]d	matches	Hello (cr -lf) World across two lines
*[0-9]	matches	234907, 5795 (or an empty string)
+[niewW]	matches	one or more strings such as Win, new, win
[a-z]	matches	any lower case string if case sensitive is on and any words if not case sensitive

Note: When [] is combined with a numeric range, the ***** operator, and the *%n>>* or *%n>starting value>* replacement operators, a search expression such as *Windows *[0-9]* would be part of a Regular Expression Counter Operation.

( ) **Subexpression Operator:** Parentheses are used to denote one or more sub-expressions. This is usually combined with the OR operator - **|**. For example,

Win(95|dows 95)	matches	Windows 95, Win 95

+n **Column Specifier:** This is used to denotes the number of columns to match either before or after an expression. The Column Specifier may be used with a simple search term such as the expression +4The or in combination with other the [] or () sub-expression operators. A range of columns to match may also be specified, such as The+4-10. Note that you should combine the + operator with [] or () if you want to be clear about literal strings that serve as an anchor to the expression. Some examples:

w+2[a-z]	matches	Wor in Hello World
+4[]w	matches	llo W in Hello World
	matches	
[ ]+5-15[0-9.]	matches	100.01, 123.9, & 543.21 in Data1 100.01 Somethin' Dat2 123.9 Nuthin' Dataa3 543.21 and junk

Text File Search and Replace - Regular Expression Replacement Operators

Notes:

- The following characters must be preceded by a \ character to be interpreted as a literal in a replace string using regular expressions: % \ < >
- We assume that Update Toolkit users are advanced computer users who are well aware of the hazards of making replaces in proprietary and/or binary files such as Word .doc's, Spreadsheets, pdf files, Access databases, and so on. If you make replaces in these types of files do so at your own risk. Experiment carefully in advance.
- Up to 24 parameters may be used at once by referring to those over number nine by moving up the ASCII table, e.g., 123456789:;<=>?@ABCDEFGH. However, if your procedure involves a large number of parameters, consider whether it might not be handled more simply using a multi-step search-replace.

Regular Expression Replacement Operators

%n **Basic Replacement Operator:** Core replacement operators use a %n convention, where n corresponds to a component in the regular expression search string. For example, %1 refers to the first expression value in the search string, %2 refers to the second, and so on. The %n parameters may be used several times, omitted, or used in any order. Up to 24 parameters may be used at once by referring to those over number %9 moving up the ASCII table, e.g., 123456789:;<=>?@ABCDEFGH. However, if your search-replace involves a large number of parameters you may find it easier to use a multi-step search-replace. Some examples of %n usage are:

Given the series: <a href="http://wait.com">S1</a>
<a href="http://waitalot.com">S2</a>
Search Expression: <a href="http://wait*[]">*[]<
Replacement Expression: <a href="http://wait%1">(Slow) %2 <
Results: <a href="http://wait.com">(Slow) S1 </a>
<a href="http://waitalot.com">(Slow) S2 </a>
Values: %1 = *[]
%2 = *[]

Given the series: <BR>
<BR>
(TAB)<BR>
Search Expression: ^<BR>
Replacement Expression: <BR>
Results: <BR>
<BR>
<BR>
Values: Here, the absence of %1 in the replace expression is used to eliminate spaces & tabs before <BR>.

Given the series: #include [stdafx.h]
#include [my_include.hpp]
#include [sr32.h]
Search Expression: ?include (<|\[)+[a-z0-9_].h*(p)+[\]>]
Replacement Expression: %1exclude [%3>.H%4>]
Results: #exclude [STDAFX.H]
#exclude [MY_INCLUDE.HPP]
#exclude [SR32.H]
Values: %1 - ?
%2 - (<|\[)
%3 - +[a-z0-9_]
%4 - *(p)
%5 - +[\]>]

< **Lower Case Operator:** This is to be used in conjunction with %n, where %n< replaces the original matched expression with the lower case version. For example,

Given:	MAKE THIS ALL LOWER CASE
Search Expression:	+[A-Z]
Replacement Expression:	%1<
Results:	make this all lower case

> **Upper Case Operator:** This is to be used in conjunction with %n: For example, %1> replaces the original first matched expression with the upper case version. For example,

Given the series:	All words beginning with letter w.
Search Expression:	w+[a-z]
Replacement Expression:	W%1>
Results:	All WORDS beginning WITH letter w.

%n>>
%n<< **Plain Counter Operator:** This is a special case of the %n>#> and %n<#< operators below, where no starting value is defined by the user. When used in conjunction with a numeric regular expression search such as *[0-9], %n>> and %n<< begin incrementing up or down with a value of +/-1 from the value of the first number found by *[0-9]. For example,

Given the series:	page5.htm, page2.htm, page4.htm
Search Expression:	page*[0-9]
Replacement Expression:	page%1>>
Results:	page6.htm, page7.htm, page8.htm

See Counter Operators for more information & examples.

%n>#>
%n<#< **Start Value Counter Operator:** An extension of the above, this operator allows you to specify a starting value for an incrementing replacement counter. *%n>starting value>* and *%n<starting value<* begin counting up or down with a value of +/-1 from the starting value you supply. This counter operator respects the number of digit places you supply. To begin incrementing upward with a value of 1, use the expression %n>0>. The expression %n>000> would begin replacements with a value of 001. Another example is:

Given the series:	Var19, Var82, Var8
Search Expression:	Var*[0-9]
Replacement Expression:	Var%1<100<
Results:	Var99, Var98, Var97

See Counter Operators for more information & examples.

Text File Search and Replace - Special Replace Operators

The following special replacement operators are available:

Operator	Function
%%srfound%%	Returns the current search string.
%%srpath%%	Returns the path of the current file
%%srfile%%	Returns the file name
%%srfiledate%%	Returns the current date stamp of the file. This is the date prior to a replace made by Search and Replace.
%%srfiletime%%	Returns the current time stamp of the file. This is the time prior to a replace made by Search and Replace.
%%srfilesize%%	Returns the file size prior to the replace made by Search and Replace.
%%srdate%%	Returns the current date of the machine.
%%srtime%%	Returns the current time of the machine.

Notes:

- The above are replace operators, not search operators.
- The above may be used in ordinary search & replace operations, regular expressions, or binary search/replace.
- It is not necessary to use Regular Expression mode to use the above operators.
- A special binary search operator, %%srformat%%=xx, which is used to Reformat Files at the specified column, xx.

Special Replacement Operator Examples

Ordinary Search/Replace

File Searched:	D:\Example\Test.txt
Initial String in File:	Page No.
Search String:	Page No.
Replace String:	%%srpath%%%%srfile%% Page No.
Results:	D:\Example\Test.txt Page No.

Complex Search/Replace

Files Searched:	home.htm; index.html
File Mask:	*.htm*
Initial String in Files:	Last Updated: 10/10/97 and Last Updated 10/12/97
Regular Expression Search String:	Last Updated: *[0-9]/*[0-9]/*[0-9]
Binary Mode Replace String:	Last Updated %1/%2/%3\r\nUrl: %%srfile%%
Results:	Last Updated: 10/10/97 Url: home.htm and Last Updated: 10/12/97 Url: index.html

Text File Search and Replace - Regular Expression Literals

The following characters must be preceded by a \ character to be interpreted as a literals in your regular expression search/replace:

Search Terms: - + * ? ( ) [ ] \ | $ ^ !

Replace Terms: % \ < >

Examples:

Goal: Find (1.1MB), (340K), (self-install)
Search Term: \(*\)
Explanation: If you don't preface the (and) the program will interpret the search and using the () paren regular expression sub-expression term.

Goal: Page references in html code that begin with the letter D and change them to lower case.
Search term: .htm*">D*</a>
Replace Term: .htm%1"d%2<\</a>
Explanation: Because the literal < character in the ending </a> tag is adjacent to your %2 operator, the program will not know if it is to part of a counter or maybe the case change operator.

Text File Search and Replace - Regular Expression Not Operator Notes

Here are some general notes about the Regular Expression 'not' operator, !.

- 'Not' expressions can be thought of a being divided into two parts - an 'expression to search for' component and one or more 'not' component(s). The '! part' always goes to the right of the expression to search for.
- The 'expression to search for' component must include some type of regular expression search operator. This is because the 'not' component needs some kind of regular expression material to consider when evaluating whether the overall expression is satisfied.
- To qualify the above, the 'expression to search for' part could be a non-regular expression term but if you search for 'something' but not 'part of something' you would end up with no hits. There is a further qualification to this… see below.
- 'Not' components must be surrounded by paren characters. A common mistake can be seen with the 'or' operator, |. 'or' itself must be surrounded by paren's. Therefore, if you use | in a not expression, double the paren's. For example, _cat*!(hat|dog)_ would not work. Use _cat*!((hat|dog))_ instead.
- Consider the search expression, _xref?!(=)_, where you want to find all instances of the string 'xref' except if followed by the = character. In words, this can be described as - "Find xref followed by anything. If the 'anything' that is returned by the ? operator is the = character, reject that search hit".
- It is OK to use multiple !( ) components. For example, cat*!(hat)!(dog) is also OK.
- Consider the search expression, _img src="*.gif*!(images/)!(alt="+[]")_. In words, this means "find img scr=", followed by anything, followed by .gif, followed by anything all the way to the end of the line. If either of the 'anything' text returned by the *'s contains the string 'images/' or alt=" " that not empty, then reject the hit. This string would find _<img src="earth6.gif" alt=""…_ but not _<img src="images/earth6.gif" alt="Graphic pic"…_ or _<img src="earth6.gif" alt=" (1019 bytes)"…_. Note that in the search expression, alt="*[]" would not work. * is the 'zero or more' operator -- it will always be true if any type of alt=" " is present.
- As modeled above, it is OK to use other regular expression search operators inside your **!** operators. Another example: ^_*!(^t)!(</p>$)_. In words, this means "find anything from the start of the line to the end of a ling, except if the first character on the line is the letter t or if </p> is at the very end of the line.
- When using **!** with replacements, don't forget to specify the search expression literals in your replacement expression. For example, given the search expression _?at!((b|c)at)_ in a replacement to put quote characters around all strings ending in 'at' (except if they begin with 'b' or 'c'), the replacement expression would be "_%1at"_. The 'at' string must be specified in the replacement expression because that is a literal in the search expression.
- Null **!** expressions can be employed as a type of "and" operator by specifying !() alone. For example, _div*!() (center|right)_ will work. In words this means, "Find div followed by anything. If there is 'not nothing' in the * part, then check for (center or right). If either exist, then return the hit.
- Another example: _*void*!(end)!()printing_ . This expression finds 'virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);' but not 'virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);'.
- The components of the expression following the **!** operator cannot be accessed via an %n replacement operator. For example, in the expression _div*!()(center|right)_ above, there would be no %2 operator for use in a replace expression.

Text File Search and Replace - Regular Expression Counters

Text Search -Replace Regular Expression Counter operations allow you to revise a sequence of numbers in one or more files. You can also insert sequential numbers in strings where no numbers exist originally. Counter operations use the _?[0-9]_, _+[0-9],_ or _*[0-9]_ regular expression search operators and either the _%n>user defined starting value>_ or _%n<user defined starting value<_ regular expression replacement operators. Counter increment +/-1 from the starting value you supply. If no value is supplied, as in the case of _%n>>_ or _%n<<_, then incrementing begins with a value of +/- 1 from the value found by your _*[0-9]_ expression. For example, if the value found by the _*[0-9]_ search expression is 10, _%n>>_ will begin at 11.

The _%n>user defined starting value>_ and _%n<user defined starting value<_ replacement operators respect the number of digit places supplied. For example, the expression _%1>1>_ begins incrementing with 2. The expression _%1>000>_ begins incrementing with 001.

Negative values can be used for the _user defined starting value_. For example, _%1>-5>_ inserts -4, -3, -2, -1, 0, … .

Arithmetic calculations are not available for counter operations. The _?[0-9]_, _+[0-9],_ or _*[0-9]_ operators do not perform numeric evaluations. These are ordinary [ ] range operators that search for the characters 0 to 9. Keep this in mind if you are searching for number strings along with $ or . characters. A search for _+[\$.0-9]_ finds '$53.00' but the program does not interpret this as 'fifty-three dollars'. Counters will not work here. The - (minus) character is an exception. _+[\-0-9]_ is OK for counter operations involving search hits such as '-53'. If you need to do searching-counters for strings like '$53.00', consider a search string such as _\$+[0-9].+[0-9]_ and (perhaps) a replace such as _$%1>>.%2_.

Incrementing counter operations may be combined with other regular expression search & replace operators. For example, a search expression such as _(file|variable)*[0-9]_ with a counter replacement expression such as _%1%2>100>_ is legal.

_%n>>_ and _%n<<_ replacements do not reset with each file. Values advance sequentially across the files found by your search criteria. The the overall sequencing is determined by the order of files in the Search Results list. This can be adjusted via the Sort File Names= switch. For example, given FILE1.TXT and AFILE.TXT, a sort order of File Name-Ascending causes AFILE.TXT to begin the sequence. A sort order of File Name-Descending causes FILE1.TXT to begin the sequence. See the second to last example below.

%n>user defined starting value> and _%n<user defined starting value<_ replacements do reset with each file. The sort order of the search results list has no effect on overall sequencing since values start anew for each file. See example below.

Regular Expression Counter Examples

Initial Contents:	Windows 98 will be released in 5 days.
Search String:	*[0-9]
Replace String:	%1>>
Results:	Windows 99 will be released 100 days.
Initial Contents:	file.htm, file.htm, ffillee.htm
Search String:	e*[0-9].htm
Replace String:	e%1>>.htm
Results:	file2.htm, file3.htm, ffillee4.htm
Initial Contents:	Var22 Var20 Var86 Var30
Search String:	Var*[0-9]
Replace String:	Var%1<50<
Results:	Var49 Var48 Var47 Var46
Initial Contents:	Var22 Var20 Var86 Var30
Search String:	Var*[0-9]
Replace String:	Var%1>00>
Results:	Var01 Var02 Var03 Var04
Initial Contents:	VarA101 VarB12 VarC0 VarA102 VarB45
Search String:	Var[a-z]*[0-9]
Replace String:	Var%1%2>08>
Results:	VarA09 VarB10 VarC11 VarA12 VarB13
File1 Initial Contents:	cat & dog cat & dog
File2 Initial Contents:	cat & dog cat & dog

Search String:	cat*[0-9] \& dog*[0-9]
Replace String:	cat%1>> & dog%2>>

Sort Order:	Name-Ascending
File1 Results:	cat2 & dog2 cat3 & dog3
File2 Results:	cat4 & dog4 cat5 & dog5

Sort Order:	Name-Descending
File1 Results:	cat4 & dog4 cat5 & dog5
File2 Results:	cat2 & dog2 cat3 & dog3

File1 Initial Contents:	cat & dog cat & dog
File2 Initial Contents:	cat & dog cat & dog
Search String:	cat*[0-9] \& dog*[0-9]
Replace String:	cat%1>00> & dog%2>00>
File1 Results:	cat01 & dog01 cat02 & dog02
File2 Results:	cat01 & dog01 cat02 & dog02

Update Toolkit Shareware

Thank you for evaluating the shareware version of Update Toolkit. The shareware license is for a 30 day evaluation period from when you first install the program. There are no functional limitations in the shareware version other than the time out.

Regarding customer support for the shareware version - If you have a question that is not answered in the documentation, please see contact us by E-MAIL at support@funduc.com and we will do our best to answer your questions. However, please be aware that ongoing support necessitates that you register Update Toolkit.

Please see Purchase & Order Information for how to obtain a registered version of Update Toolkit.

License Information

Update Toolkit

See Ordering Information for information on how to purchase (register) Update Toolkit.

License Agreement and Warranty Disclaimer

You should carefully read the following terms and conditions before using this software. Use of this software indicates your acceptance of these terms and conditions. If you do not agree with them, do not use the software.

Shareware Version License

The shareware version is not free software. If you are using the shareware version you are hereby licensed to use this software for evaluation purposes without charge for a period of 30 days. If you use this software after the 30 day evaluation period a registration fee is due (see Ordering Information).

If you are using the Shareware Version of the software you are also hereby licensed to: Make as many copies of the Shareware version of this software and documentation as you wish; Give exact copies of the original Shareware version to anyone; Distribute the Shareware version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission.

Unregistered use of Update Toolkit after the 60-day evaluation period is in violation of federal copyright laws.

You may not decompile, disassemble, or reverse engineer Update Toolkit (shareware) in any manner.

Registered Version License

One registered copy of Update Toolkit may be dedicated to a single person who uses the software on one or more computers or workstations that are for their own exclusive use and for which they are the sole user.

A single user license may be used to process as many files on as many drives or network shares as you like. However, a single user license registration may not be made available in any form to multiple users or workstations.

Licensing is not based on "concurrent" users. If multiple workstations are to access the program in any form and for any length of time you must purchase licenses for the number of users that will access the program. If you make the registered version of Update Toolkit available through a network and accessible by multiple users or workstations you are required to purchase licenses for the software covering all workstations that will access the software through the network. Special pricing is available for multiple licenses. See Ordering Information for information on how to purchase (register) Update Toolkit.

You may not decompile, disassemble, or reverse engineer Update Toolkit (registered) in any manner.

Workstation Update And/Or Special Use License

Special licensing (and pricing) is available for modifying or searching multiple workstations, directly or indirectly, via scripts, command line processing, or logon routines. The purchase of a special use license allows you to use the program to update the number of workstations for which you purchased special use licenses for. Unless otherwise noted at the time of your purchase, a workstation update/special use license specifically does not allow individual workstation users to access the program for interactive use. Do not create desktop shortcuts or launchers that allow workstation users to access the program outside of the specific update routine for which you purchased the special use license.

Please contact sales@funduc.com for more information on special use licensing.

Ownership Rights

Funduc Software, Inc. owns and retains all copyrights, patents, trade secret rights, trademarks and intellectual property rights to this software. Registration &/or use of the software does not grant you any right to the intellectual property in the product.

Customer Support

Customer support can be obtained by visiting our homepage at www.funduc.com or www.searchandreplace.com by E-MAIL (support@funduc.com).

Restrictions

You are prohibited by law from selling, renting, leasing, loaning, or donating the program or documentation in whole or in part without specific advance agreement between yourself and Funduc Software, Inc. You are prohibited by law from redistributing the licensed version of the program by any method or in any form except to the specific individuals for whom the software was purchased. You are prohibited from decompiling, reverse engineering, disassembling, or otherwise reducing either shareware or licenses versions of the program to human readable form for any purposes whatsoever. You may not modify the program or documentation in any manner. You are prohibited from permitting anyone else from undertaking any of the above actions.

Governing Law

This agreement shall be governed by the laws of the State of Michigan, United States.

Disclaimer of Warranty

This software and the accompanying files are sold "as is" and without warranties as to performance of merchantability or any other warranties whether expressed or implied.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program. Any liability of the seller will be limited exclusively to product replacement or refund of purchase price.

Customer Support Information

If you have a question that is not answered in the documentation, please try our Web site at www.funduc.com for any FAQ (Frequently Asked Questions) documents that may become available for Update Toolkit. If you need further support, contact us directly by E-MAIL at support@funduc.com. Please give as much detail as possible about the operation you want to perform and what the difficulty is.

To report a bug, E-MAIL: bugs@funduc.com.

Registered users are eligible for support and updates as they become available.

Shareware Users: We will do our best to answer your questions, however, please understand that ongoing support necessitates that you register Update Toolkit.

We welcome suggestions for improving Update Toolkit. If you have an idea, first check the documentation and README.TXT file to make sure they are not already implemented or planned. If not, drop us a note at support@funduc.com.

For ordering information, please see Ordering Information.

Thank you for using Funduc Software products!

Acknowledgements

Thanks to you, our customers, for your suggestions & support!

Update Toolkit

http://www.funduc.com

See Ordering Information for details on how to purchase Update Toolkit or write us at info@funduc.com. See License Information for details about licensing. For customer support, contact support@funduc.com or visit our web site at www.funduc.com for FAQs & program updates. Also check our web site for product information, trialware & freeware downloads of our other award winning software products.

Funduc Software - Purchasing & Ordering Update Toolkit

The base price of Update Toolkit begins at $50/single user, plus $5 to $8 S&H if you want a diskette sent by US or International mail. Email delivery is free. Quantity discounts are listed in the fax/mail order form and our web site, http://ww.funduc.com.

Purchase Orders? Please see our online PO Policy PO Policy. before submitting a PO. If you cannot reach our web site, write support@funduc.com or call 734 462-1552.

As part of the registration, you will also receive free updates and customer support (by Email) for one year. Updates are sent to registered as they become available. They incorporate your suggestions, so please send them! If you have problems receiving the program or need customer support, contact us directly by Email at support@funduc.com.

Credit Card Registration:

- **Secure Internet Transaction:** If you have a secure Web browser such as Netscape or Internet Explorer, you may register by credit card via a secure Web transaction from our web site at: http://www.funduc.com.
- **Fax Registration:** To register using a credit card via fax, open and print the fax/mail order form. You may also edit and print the file, 'Order.txt', in the program subdirectory. Please write clearly or type when you complete the form. Please check our web site - www.funduc.com/contact.htm - for the fax number to use. Fax orders are processed within a few hours during the USA business day. Allow 7-10 days for postal deliveries. Write support@funduc.com if you have a question about your order.
- **Phone Registration:** Call the number indicated on the page at www.funduc.com/contact.htm to order using a credit card. Please specify that the order is for Duplicate File Finder. Phone orders with email delivery are typically processed within one business day. Allow 7-10 days for postal deliveries. Write support@funduc.com if you have a question about your order.

Ordering by check: To register by check, open and print the fax/mail order fax/mail order form. You may also edit and print the file, 'Order.txt', in the program subdirectory. Please write clearly or type when you complete the form. Mail the form and a check to the address indicated on the form. Please allow 7-10 days for check order processing and then contact support@funduc.com if you did not receive the program. Payments must be in US dollars drawn on a US bank or international postal money order in US dollars. If you send a postal money order, do so only by some form of registered mail that can be traced!

Update Toolkit Fax & Mail Order Form

Funduc Software, Inc. - Fax and Mail Order Form
Update Toolkit

Funduc Part Number: 501-UT-4881

Fax Orders: Fax to the number indicated on the page at www.funduc.com/contact.htm

Check Orders:
Funduc Software, Inc.
P.O. Box 530904
Livonia, MI 48153-0904
USA

-------- PRINT NEATLY OR TYPE! --------

Price Schedule	Price ($US) Per User	Number of Users	Price
Single user	$50.00	_______	$________
2 to 9 users	$40.00	_______	$________
10 to 24 users	$30.00	_______	$________
25 to 49 users	$25.00	_______	$________
50 to 99 users	$20.00	_______	$________
100 to 499 users	$15.00	_______	$________
500 to 999 users	$10.00	_______	$________
1000 to 4999 users	$5.00	_______	$________
5000 to 9999 users	$3.00	_______	$________
10,000+ users	$1.50	_______	$________
	Residents of Michigan, United States	- add 6% sales tax	$________
	Preferred Shipping Method	Mail 3.5 disk (US) $5 S&H	$________
		Mail 3.5 disk (overseas) $8 S&H	$________
		Email (Free)	$___ 0 __
		TOTAL PAYMENT	$________

Credit Card Information

Card Type (Check One): Visa[ ] American Express[ ] MasterCard[ ] Discover[ ]

Name as it appears on card: ________________________________

Credit Card Number: ________________________________

Expiration Date: ____________

Street Address: ________________________________

Street Address: ________________________________

City, State Zip: ________________________________

Country: ________________

Your Information (Write 'Same' if not different from above)

Your Name: ________________________________

Company: ________________________________

Street Address: ________________________________

Street Address: ________________________________

City, State, Zip: ________________________________

Country: ________________________________

Day Phone Number: ________________

Fax Number: ________________

Alt Phone Number: ________________

* EMail Address:* ________________________________

Signature: ________________________________

Date: ________________

* Include E-Mail address if you want email delivery.

Comments: