

WinSince: A file management automation tool

Overview

WinSince is a **unique and powerful file manager** that can help you find and work with your files. Unlike most file managers, WinSince is **list oriented**: you search for files you are interested in and build a list of those files. Once you have a file list, you can **perform any action** you need on the list, including DOS commands, DOS programs and Windows programs.

For example, you might want to find all *.BAK files on your hard drive and delete them. Other file managers require you to set up such a file search each time. Not WinSince! With WinSince you **set up an icon** just once that will **automatically perform that file task** any time you need it. You can run your action in a Window or minimized and out of the way.

By showing the files you have selected as a list, WinSince makes file management a lot like using a grocery list. You can **sort the list of files** in lots of ways, such as by file name, extension, date and size. You can **add files** to the list by searching for a different type of file, or searching in a different directory. You can also **remove files** from the list, and **select only certain files** to work on.

One of the strengths of WinSince is its ability to **search for files created since a particular date**. In fact, that is how it got its name. As with many file managers, you can tell WinSince the date of the files you want to find. But WinSince also allows **relative date searching**, such as finding files created some amount of time before now.

For example, you might want to find all report files created in the last 8 hours and copy them to another directory for safe keeping. With WinSince this is easy. You can also tell it to only **look for files that have not yet been backed up**, which is a good safety net for those times between backups when you aren't exactly sure which files have been worked on.

If you want, WinSince will also create a text file that **logs all the files it finds** and the actions it takes. This makes WinSince really handy for keeping track of **which files have changed**. You can have it add new information to an existing log file or even create a new one each time you run WinSince. This feature is especially useful for people using networks or working together on files, since it provides a kind of audit trail.

WinSince also has a **built in timer** which you can use to **perform repetitive or time dependent activities**. Using the timer you could set up your own timed backups for programs that don't have that feature. For example, you could have a copy of WinSince whose job it is every hour to search for your type of spreadsheet file, check if any have changed, and copy them to another directory. You can have as **many copies of WinSince running** as you have tasks for it to do.

WinSince can also **search files for text**. This can be very useful for finding word processing and spreadsheet files. Its **Search History** shows how many files were found and searched. It can also copy file information from the File List to the Windows Clipboard for moving to other applications.

WinSince has been described as being more than a utility, but rather being a **utility construction set**. Its customizable DOS and Windows Actions and definable File Type buttons make it very **flexible**, while its list approach makes it **easy** to use. Give it a try. I think you'll find, like me, that you have lots of file-related tasks that can be easily automated, to save you time and effort.

Version 5.0 of WinSince adds an **Action Button Bar** to the 10 definable Action buttons. This gives more room for defining your own Actions, which can include the Button Bar Actions. You can also **print the file details for selected files**, and **print the list of Actions**. WinSince 5.0 also has expanded its storage size and can now **handle an entire hard drive** at a time (up to 7000 files). The new **VCR control** on the File List makes paging through long file lists easy. The feature history in the Readme.txt file has more details on the evolution of WinSince capabilities.

Rob Hueniken, March 1994

Help Topics available:

- [Quick Start](#)
- [Using File Buttons](#)
- [Selecting the Date](#)
- [Editing and Using Action Buttons](#)
- [Double Clicking to start a program](#)
- [Options menu](#)
- [Hints and Examples](#)
- [Using Command Line Qualifiers](#)
- [Copying and Deleting Files](#)
- [Actions in the Distribution Kit](#)
- [Drag and Drop Capabilities](#)
- [Setting File Attributes](#)
- [Using the Timer](#)
- [Searching for Text](#)
- [Paging Mode and VCR Controls](#)

Quick Start

This section gives a quick tutorial on the major WinSince features. It is intended for first time users. WinSince uses normal Windows conventions, so previous experience with moving between directories and selecting files from a list will be useful. See the other Help sections for detailed help.

For the tutorial flow to be correct, follow the steps right after starting WinSince. The tutorial will suggest what to click the mouse on at a particular time.

A Quick Tour of WinSince

The main WinSince Screen

When WinSince starts the current drive and directory will be shown in the top right of the screen. You can change to a new directory by double clicking on a subdirectory name.

Beneath the directory boxes is the **File Type** button section, which has 12 definable buttons. Click on the one that has the caption ***.exe**. The **Directory total** file list (middle top of the screen) will show the number of .exe files in the current directory. Click on the ***.*** button and see how many files there are in total.

The date that WinSince will use while searching is the **Find Since** date, shown on the middle left of the screen. Right now it will show today's date and the word **Today** to the right. Under the Find Since date are the VCR-like controls to change the date. Click on the **"<"** button to move back a day to Yesterday. Press the **Today** button to return to Today's date. This is fun ! The time defaults to midnight, but you can set it to any time, for very accurate file checking.

The big button above the colorful icon in the middle of the screen says **Find Files** and is used to start the search for files. Click on the **Find Files** button. WinSince will fill the **File List** box in the top left of the screen with any files changed today (because the **File Type** is ***.*** and the **Find Since** date is Today). If the **File List** box is empty then no files were changed in this directory today.

Click on the **Clear Date** button: the date box will now say **No Since Date**, which tells WinSince that it should not worry about the file creation date when searching for files. Click on the **Find Files** button. This time there will likely be files in the top left list box (unless your directory is empty, in which case you need to double click your way to a busier directory) and it will have a title that begins with **No date selected**.

The file details shown are: the **name** of the file, its **creation date** (in MM-DD-YYYY HH:MM:SS format), its **size** in bytes, any **attributes** (such as **a** for **archive**, **r** for **read-only**), and the **full path** of the file.

Click on any one of the files. It will show in reverse video. Click on the **Dir** button found at the bottom of the screen. The **Dir** button is one of the 10 definable Action keys. You will see a DOS box appear and the DOS Dir command will be run on the file you selected. After the command is completed, you will see a **Press a key when ready...** request in the DOS box; this pause is built into the **Dir** Action definition, which you will see soon. Press return to complete the Action.

Next, look at the section above the Action buttons on the left. This section lets you choose whether you will search for files in the selected **Directory**, down into **Subdirectories**, or across entire **Drives**.

When you click or double click on the **Drives** radio button a listing of the available drives is shown, such as **a: b: c:**. Click on the box to the left of your hard drive's letter, then on the **OK** button to accept the selected drive. If you click on **Cancel** then the file search will reset to the previous type of file search, such as Directory or Subdirectories. Next click on the **Free Bytes** button on the lower right of the screen. In the box below you will see the number of bytes free on this disk and the percent free. You can select **multiple drives** and click on the **Free Bytes** button to give you a total. This can be very handy for **checking total disk space available**.

The **Set Type** button is found on the right beneath the **File Type** buttons. Click on the **Set Type** button. A small form will overlay the File Type buttons. Click on one or more of the small buttons, and watch the button's file type change to the current **File Type to Set (*.*)**. This is fun ! If you make a mistake then press the **Reset** button to restore the values. Pressing the **Cancel** button will exit the button-setting form without changes; pressing the **OK** button will change the values on the main WinSince screen. Changed **File Type** buttons can be saved from the **File** menu or when you **Exit** the program (bottom right of main WinSince screen).

The Edit Screen

The last major section to examine is the **Edit** menu item. Click on the **Edit** item along the top of the main WinSince screen, followed by the **Edit Actions** menu item. The **Define Action Keys for WinSince** editing form will appear.

There are two areas in the Edit screen. At the top is a list of **summarized Actions** and at the bottom is the actual editing area. Click on the **Action 7 Dir** item in the top section. It will change to reverse video, and the lower editing area will change to reflect the components of this Action.

For the **Dir** command you will see that the **Action to Do** is **Dir %a**. The %a refers to the full path and file name. Other file tokens include %d for the drive, %p for the path and %f for the file name. When the action is actually run on a selected file WinSince will replace the %a place holder to yield a real directory command, such as **Dir c:\windows\win.ini** (drive=%d=c, path=%p=windows and file=%f=win.ini).

Notice that the **Windows or DOS** check box is empty because the **Dir** command is a DOS command, not a Windows program. When you start defining your own Actions, they can be DOS commands and programs, or Windows programs.

By setting the various fields you can modify the selected Action to do just about anything. A more thorough explanation of Action editing can be found in the **Editing and Using Action Buttons** help section. Examples of Actions can be found in the **Actions in the Distribution Kit** section.

Click on the **Exit** button in the Edit screen. This will return you to the main WinSince screen.

To exit the WinSince program, click on the **Exit** button in the bottom right of the main WinSince screen.

Getting Fancy

This Quick Start section describes the basics of using WinSince. For a full understanding of its capabilities, read the other sections. In there you will learn how to:

- Drag and Drop files from the directory into the File List and onto Actions,
- Set the search date by dragging and dropping a file onto the date field,
- Remove files from the File List,
- Create a log file to audit which files you have found and worked on,
- Customize the files searched for when WinSince starts,
- Turn off the warning when you clear the File List,
- Sort the File List by Name, Size, Date, Path, Extension or Attributes,
- Use the Timer to automate file management,
- Search for Text in files,
- Copy file information to the Clipboard,
- Use the Button Bar Actions,
- Page easily through longer lists of files,
- Print a list of the Actions,
- And lots more.

When WinSince starts, it defaults to finding all files changed today in the current directory. Above the main file list box (top left) will be shown the number of files found, the date being checked, and the type of search.

You can use the **Add Files** control immediately above the Find Files button to add to the File List with each file search. The File List will then not be cleared when you click on **Find Files**. File names are added to the File List if they are not already in the File List. This control is useful for creating lists of many types of files or files from many different directory areas. It is equivalent to the **/Add** command line option.

There are 3 file search options you can select, depicted as radio buttons on the main form:

1) Search selected **Directory**: This is the default mode, searching only the directory selected using the drive and directory selection controls.

2) Directory and **Subdirectories**: This begins the file search in the current directory and includes any subdirectories under the current directory.

3) Search entire **Drives**: This option allows searching of one or more entire drives. These drive selectors are shared with the **Free Bytes** function. When you click on the Drives radio button, a form pops up to allow you to select one or more drives. If you **Cancel** then the previous search type is restored, such as Directory or Subdirectories. If the Drives radio button is already selected, you can **select other drives by double clicking** on the Drives radio button.

The drive and subdirectory selection controls operate in the standard Windows manner. The "**Directory total**" file list box shows the names of files matching the File Type in the current directory, irrespective of the Date selected. Single clicking on a file in this list will show the file details of this single file below the main file details list box.

Under the subdirectory control is the **File Type** text box. You can directly enter the file type here, such as HELLO.* or you can click on one of the 12 file type buttons. You can use the * (multiple character wild card) and ? (single character wild card) symbols as accepted by DOS. You can also request multiple file types to be found if you separate the file types with a semicolon. For example, you could set **File Type** to be *.EXE;*.TXT;*.RPT

Defining your own File Type Buttons

To define your own **File Type button**, press the **Set Type** button. A small form ("Set File Button") will overlay the existing file buttons. You can edit the "**File Type to Set**" file name as needed. To set a button to the **File Type to Set** file name just click on one of the 12 buttons. It's fun! Press the **OK** button to accept the new file buttons, or **Cancel** to return without changes. The **Reset** button will reset the file button values to those when you entered the form.

You can make your File Type button changes permanent by using the **File** menu **Save File Buttons** menu selection. You will also be warned if you are about to exit without saving new Button types.

Drag and Drop a File into the File List

You can drag and drop a file from the directory control into the File List directly, without going through the process of using the Find File button. This makes it easy to create a diverse list of files quickly. (You can also use the Add Files feature to create diverse lists). Dropped files are added to the beginning of the File List when not in paging mode, so it is easy to see what has been added. In paging mode, dropped files are added to the end of the File List for greater speed. The total size of the File List is also updated.

The **Free Bytes** button is used to determine the amount of free disk space remaining on your disk drives. Before clicking on the Free Bytes button select one or more disk drives in the Search entire Drives section. When you click on the Free Bytes button it will check all selected drives, including any floppy drives you selected. The free space in bytes and as a percent of the total available space is shown. Because Free Bytes uses the same disk drive selectors as the file search across drives, you should check which drives you have selected before pressing the Find Files button to find files.

Sorting the File List

Once you have created a list of files, you can sort the files in a variety of ways by using the **List** menu. This makes it easy to find a particular file in a large list of files, or to determine the largest file in a directory area.

The available sort methods include:

Sort by File Name	ex. WIN.COM
Sort by Full Path	ex. C:\WINDOWS\WIN.COM
Sort by Extension	ex. .COM
Sort by Creation Date	oldest files to latest files
Sort by Size	smallest files to largest files
Sort by File Attributes	ex. ar = Archive/ReadOnly

For sorts other than by File Name and Full Path, the list is sorted secondarily by File Name. This would mean that for a sort by Size, files with the same size are also sorted by File Name.

When you sort the list, any selected files are deselected. When you get a new set of files, the File List reverts to being sorted in the order in which the files were found. When you drag and drop a file from the Directory list into the File List, the file appears at the top of the File List.

Selecting the File Date

Absolute Date

Next to the "**Find Since**" prompt is the Date text box. You can directly enter the date here in **MM-DD-YYYY HH:MM:SS** format, or use the VCR controls to move forward or backward by a Day or Week at a time. The adjacent text box shows the date offset, relative to today (ex. Yesterday).

For example, to select files created since January 20, 1993 the date field would appear as 01-20-1993 00:00:00 . Note that the time defaults to midnight, so that files created any time during the selected date will be found.

For more specific searches, you can set the time in 24 hour format. For example, to find the files changed or created since noon on January 20, 1993 the date field would appear as 01-20-1993 12:00:00 . Note the single space between the date and the time portions.

The time portion remains the same when you change the date portion with the VCR Date controls. You can **reset the time to midnight** by **double clicking** anywhere in the date field.

The **Clear Date** button will let you look for files without regard to their creation date. This is very handy for finding all files of a particular type.

The **Save Date** button stores the current date for later retrieval with the **Reset Date** button. If you find a date that shows important file activity you can use the Save Date button, then change the date to look more, knowing that you can get back to the files for the previous date by pressing the Reset Date and Find Files buttons.

When WinSince starts, it defaults to Today, unless you use the /SINCE command line qualifier. For a discussion of /SINCE, see the "Using Command Line Qualifiers" section.

Relative Date

WinSince allows another type of time specifier that is very handy for finding files created some amount of time previous to Now. The format for this relative date is **-DDD:HH:MM:SS** which starts with a minus sign (-) followed by any number of days. The number of hours can be from 00 to 23, the minutes and seconds from 00 to 59.

For example, to select files created in the last hour, the date field would appear as -0:01:00:00 When you press the **Return** key WinSince will convert the relative date to an absolute date in the text field to the right of the Find Since date field (instead of saying "Today" for example).

Relative date searching is very useful for detecting recent changes to your files.

Drag and Drop the Find Since Date

You can drag and drop a file from the directory control onto the Find Since date. This sets the file search date to that of the file dropped, and is useful for checking for files created since that particular file was created.

Editing and Using the Action Buttons

In many instances, just having a list of files matching a date and file type is enough. But other times you will want to perform actions on some or all of the files found. The Actions available in WinSince can be both Windows programs and DOS functions and programs. In addition, WinSince has built in Actions to do file Copying and Deleting.

To select a single file to act on, click on the file name in the main list box. To select multiple files, hold down the Control key while you select additional files, or hold down the Shift key to select all files up to the file clicked. You can select all files using the **All** button, and unselect all files using the **None** button.

There are 10 definable Action keys located near the bottom left of the screen in two rows. Along the very bottom of the screen are the Button Bar Actions, which can not be redefined, but which can be combined in your own definable Actions. To view the current set of Actions enter the **Edit** menu option. The list box shows the descriptions of the 10 Actions, as well as the **Cut Buffer**. For a discussion of the meaning of the supplied Actions, see the help section **Actions in the Distribution Kit**.

The **Action Caption** can be up to 10 characters long.

The **Action to Do** can be up to 100 characters long. It can be a DOS command or program (such as COPY or FOX.EXE) or a Windows program (such as EXCEL.EXE). Special substitution symbols allow you to pass the selected files to the Action in very flexible ways:

%A	All	Substitutes the entire path and file name
%D	Drive	Substitutes the drive letter, excluding the ending ":\"
%P	Path	Substitutes the path, excluding the starting and ending "\"
%F	Filename	Substitutes the 8 character name, the period, and the file extension
%N	Name	Substitutes only the 8 character name, with no period
%E	Extension	Substitutes only the 3 character file extension

Example: If the file list shows an entry for **C:\WINDOWS\FILE.TXT**, and the Action is **COPY %A A:\%N_2.%E** you would get a resulting command of **COPY C:\WINDOWS\FILE.TXT A:\FILE_2.TXT**

I.e. %A is C:\WINDOWS\FILE.TXT
 %D is C (note how there is no ":\\" ; you put that in)
 %P is WINDOWS (note how there is no "\" at front or back)
 %F is FILE.TXT
 %N is FILE
 %E is TXT

There are now two additional File tokens that are used to point to the current drive and directory:

%CurDrv	Substitutes the drive letter of the current directory
%CurDir	Substitutes the current directory

If the current directory within WinSince is C:\WINDOWS\SYSTEM then %CurDrv is C and %CurDir is WINDOWS\SYSTEM . Notice how the ":" and starting and ending "\" are not included. This is the general rule when creating WinSince Actions. These two substitution symbols allow for point and click copying and deleting. Examples are found in the distribution Since.ini file.

There are also four built in WinSince functions:

%Copy	Copy a file to another location
%Del	Delete a file
%Attrib	Modify the file attributes
%Action	Combine Actions together

The first two functions are described in the help section Copying and Deleting Files. **%Attrib** is described in the section Setting File Attributes. **%Action** is described at the end of this section.

Click on an Action in the Action list box to begin editing it. When done, click on the **OK** button; the changes will then appear in the Action list box and be ready for use. The caption will also be changed on the corresponding Action button in the main WinSince window. The **Cancel** button will cancel the changes if the **OK** button has not yet been pressed. You can use the **Copy** button to copy a selected Action to the Cut Buffer. The **Paste** button will copy the contents of the Cut Buffer to a selected Action. The **Exit** button returns you to the main WinSince screen. Note that WinSince will do some checking of the Action contents when you press OK to try to determine if the Action is incomplete.

The **Preview** button is a handy way to visualize how the Action will appear when it is run on a file. It will do the file name substitutions using a sample file name to demonstrate how the Action command will appear. This is useful for verifying that you have put all of the suitable ":", "\" and other punctuation in the correct locations.

The **Print** button will print out the Action definitions for the 10 definable Actions as well as the Button Bar Actions. Having a hard copy of the Actions can make combining Actions easier.

The **Window Format** will usually be Normal, Show Minimized or Show Maximized. For %Copy and %Del Actions the format is always Normal.

The **Run Windows Program** check box should be selected to run a program requiring Windows. For DOS commands and programs leave this box unchecked.

DOS Actions are best run using the **DOS Batch File**, which uses the SINCECLS.PIF file to run the Action in a windowed DOS box, and to close the Window when done. You can use the PIF editor to modify this to use Full Screen, etc. as required. If you do not use the DOS Batch File option then each iteration of the Action on a list of files will create a new DOS box under Windows, usually causing you to run out of memory. By using the DOS Batch File, a single batch file is created (SINCE001.BAT in the TEMP directory) that needs only a single DOS box to run.

If you check **Pause between DOS Actions** then a DOS **Pause** command is inserted between each action in the batch file.

If you check **Pause at end of DOS Batch** then a DOS **Pause** command is inserted after the last action in the batch file. If you do not specify the pause at the end then the window will close when the action is completed, which is useful if you run a minimized action that you don't want to see the final status of.

Warn before Starting

The **Warn before Starting** check box will present you with a message box before the Action is performed on a selected file. If you select **No** at Action time then the next selected file will be presented. If you select **Cancel** the action is ended for all selected files. For **DOS Batch File** actions a single warning is given before the batch file is started, showing a sample action contained in the batch file. For Windows, %Copy and %Del Actions on multiple files, a "Yes to All" choice is also given at Action time, making it easy to check the Action being done for a single file then accepting all further action on the remaining files. It is recommended that the **Warn before Starting** check box be set for

all %Copy and %Del Actions to give you a chance to verify that the Action is what you intended.

For Windows actions there is no direct parallel to the PAUSE ability of the DOS batch file. Instead, you can check the Warn before Starting box. Before the action is started on the next file you will be presented with the Warning box. Ignore the Warning box and click on the Windows application. When you finish the application the Warning box will still be there. You can then progress to the next file, or cancel subsequent actions.

You can make your Action button changes permanent by using the **File** menu **Save Actions** menu selection. You will also be warned if you are about to exit without saving new Actions.

Note: if you make an error while editing an Action, exit WinSince without saving the changed Action definitions. When you restart WinSince the previous actions will be there. If you ever need to, you can get the original WinSince Actions and File Types back by deleting the Since.ini file and starting WinSince.

Run an Action on a Single File

You can drag and drop a file from the directory control onto an Action button. This is very handy for running an Action on a single file, since there is no need to go through the process of using the Find File button and selecting the file from the File List. The same Action warnings are in effect.

%Action: Combining Actions

By using the **%Action** command you can run multiple actions on the list of files with a single button click. You can also run the same combined Actions from the startup command line with the **/Action** qualifier.

The format for a combined Action is:

%Action4+%Action1+%Action11

You can add up to nine Actions together. The order in which the Actions are done is the order they appear in the Action definition. In the example above, Action 4 is done, then Action 1, and finally Action 11. To use the **%Action** command requires at least two Actions to be combined.

A **%Action** command can not point to another Action that contains a **%Action**, and it can not point to itself. In the example above, the **%Action** could not be assigned to definable Actions 4 or 1. **%Actions** can point to **definable Actions** (Actions 1 to 10) and **Button Bar Actions** (Actions 11 to 20).

In the Edit screen, the portions important to the **%Action** command are the **Action to do**, the **Action Caption**, and the **Warning before start**. If the Warning is set, then before the combined Action begins you will be warned that it is a combined Action. This is independent of the Warning for each of the combined Actions, which you can set or reset in the Edit screen. You can use the Option menu to suppress all of the warnings for Actions, as long as you are confident of their effects. When the warnings are suppressed, a **yellow caution triangle** appears on the main screen above the Actions with the Caption "**Warning off**".

Double Clicking to start a program

You can start any executable program (.exe, .com, .bat, .pif) by double clicking on its name in either the directory file list or the selected files list box. You can also start the associated program for the file's extension (such as starting NOTEPAD.EXE when you double click on README.TXT).

You can set the **Warn before run double click** option in the **Options** menu to verify what program will be run when you double click on a particular file type. Setting this option is useful for verifying the file associations within Windows. Once you are confident that clicking on a file type runs the right program you can reset this option.

Options Menu

The 10 options for WinSince are:

1) **Get file information when start up:** If set, WinSince will look for files created today in the current directory when WinSince starts. If you do not often want a list of the changed files in the start up directory then reset this option. Resetting this option may save a bit of time at start up if you start in a directory with many files. Use the **Find Files** button to search for files. If the **/search** command line qualifier is used it overrides this option to always get file information at start up.

2) **Warn before run double click:** If set, WinSince will present you with a warning box when you double click on a file name. The warning box will show the command about to be run (such as NOTEPAD.EXE README.TXT if you double click on readme.txt). Setting this option is useful for verifying the file associations within Windows. Once you are confident that clicking on a file type runs the right program you can reset this option.

3) **Warn before clear File List:** This option is set in the distribution package. Once you develop confidence in your management of the files in the File List you will likely want to turn this option off. When the option is set you get a warning before the File List is cleared from the List menu, and before entries are removed from the list with the List | Remove menu item.

4) **Run program on own when double click:** This option is usually set. If reset and you double click on a program (.exe, .com, .bat, .pif) you will pass the program name to the program itself (such as EXCEL.EXE EXCEL.EXE). This is rarely the action wanted when you double click on an executable, but it is included for use under special circumstances.

When you exit WinSince, the preceeding four options are saved. Six other options are not saved, because of their batch orientation. The following six options can be set in the Options menu, or specified on the start up command line:

5) **Suppress warning before Action (!):** This option will override the **Warn before Start** warnings defined for **all** Actions in the Edit screen. For example, you will usually have the Warning set for a %Copy or %Del action in order to let you verify that you are acting on the files you want. If you select this option then the warning will not appear. This option is not saved when WinSince exits. For careful users, or when running WinSince in batch mode on proven file situations, you may want to use the **/NoWarn** command line qualifier to turn off the warning defined in all Actions.

When you select this option, a **yellow caution triangle** appears on the main screen, above the Actions buttons, with the caption "**Warning off**". This gives you a solid, visual indicator of when warnings will not be given before an Action. **Note** that Actions can be defined to not give a warning before they start, so the absence of the yellow caution triangle could still mean that no warning will be given. Use the Edit screen to set the "Warn before start" option for any Actions you want a warning for, and **use the Suppress warning option** with care.

6) **All files selected after each search:** This option saves you from needing to click on the "**All**" button after a search. It is handy when you know you will want to select all files for a sequence of searches. Be mindful that this option, in combination with the "Suppress warning before Action" option, requires you to be careful before invoking the Delete or Copy Actions. This option is not saved when WinSince exits, but you can add **/All** to the command line.

7) **Ignore non-fatal file copying errors:** This option will prevent the warning usually shown when WinSince is unable to copy a locked or open file. It is useful when running unattended file copying in batch. Fatal errors, such as a full disk still have their messages shown. This option can be set by using the **/NoCopyWarn** command line switch.

8) **Copy only if Different date or size:** If this option is set then the source file will not be copied to the target directory unless the source file has a different creation date or size than the target file. This is useful for copying only files that are different than the ones in the target directory. It can also speed up file copying since not all selected files are copied. Note that the source file could be an older or smaller file and still be copied. Note also that a change to the file that does not change the date or size is not detected, but this is not common. If you are concerned about copying only newer files then see the next option (If Newer). You can set the option using the **/IfDifferent** command line switch. If the option is not set then files are copied regardless of size or date (unless the If Newer option is set).

9) **Copy only if Source file is Newer:** If this option is set then the source file will not be copied to the target directory unless it is newer than the same named file in the target directory. If the target directory does not have a file by that name then the copying will be done. If the file dates are the same or if the source file is older then the file will not be copied. The size of the files is not checked. You can set the option using the **/IfNewer** command line switch. If the option is not set then files are copied regardless of size or date (unless the If Different option is set).

Interaction of Copy if Different and Copy if Newer

The **If Different** option is tested first, and passes on files that are different in size or date. If both options are set then the only files copied are newer, and they may have a different size. In other words it is the same as just setting the **If Newer** option.

10) **Ignore Attribute Setting Errors:** If this option is set then WinSince will not warn you when it was unable to set the file attributes with an Action containing %ATTRIB. This option is useful for when you know that certain files will be locked or opened by another program and are not concerned that their file attributes are not being changed. You can set the option using the **/NoAttribWarn** command line switch.

Hints and Examples

By customizing the Action buttons you can create a very powerful and easy to use working environment that can solve virtually any task involving file manipulations and program execution.

The copy of **Since.ini** provided with the program provides additional examples of defining the Action buttons. See the help section **Actions in the Distribution Kit**.

Example: Save today's work to floppy (into a single directory)

One of the reasons I wrote WinSince was to be able to see which files had been changed during today's work session and to copy them to a floppy for safe keeping. By selecting a date of **Today**, searching for files, and using an Action defined as **%COPY %A A:\%F** I can now save today's work to floppy quickly and easily. Notice how you have to put the \ in. Because there is no %P in the file to copy to, all of the files go into a single directory. This is the "**Copy to A:**" Action on the distribution disk.

Example: Save today's work to floppy (into a directory structure that looks just like that on the source hard drive):

Select a Since date of **Today**, search for some files in the subdirectories, select the files you want (perhaps All or All with some removed) and use an Action defined as **%COPY %A A:\%P\F** The %COPY command handles the creation of the subdirectories as needed. WinSince will verify that there is sufficient disk space to copy the files. You can also use the **Free Bytes** button after the search to check that the target floppy has enough space to copy the files to. The total size of the files is listed at the bottom of the file list. The size of the selected files can be shown by using the **List** menu.

Example: Save today's work to floppy and move it to another machine:

This is like the previous example, except there are two Actions done, one on each machine. Use the previous example to copy files to the floppy. Insert the floppy in the target machine, search for all files in the floppy subdirectories, then use an Action defined as **%COPY %A C:\%P\F** Here I am assuming that A: is a floppy and C: the target hard drive. Notice how %A specifies the entire path and file name, and has nothing to do with drive A:. As an alternative, after you get the list of files, you could point to the A:\ or C:\ root directory and use the **Copy To** action in the distribution kit. This saves needing to define a special button.

Example: Save a copy of files into a directory structure that looks similar to the original, but indented by a subdirectory layer:

Search for some files, select the files you want and use **%COPY %A %D:\SAVE\%P\F** Essentially, this is prefacing the original Path with another directory. This is the "**Save Copy**" Action on the distribution disk. Similarly, you could do **%COPY %A G:\SAVE\MORE\%P\F** to move files even deeper on another drive (G:).

Example: Get a copy of files existing in one directory from a different directory

Suppose you have a directory called C:\FILES with 5 files in it, and another directory C:\NEWFILES with 10 files. The files in C:\FILES need to be set to the contents of C:\NEWFILES but you don't want all 10 files, just the 5 files. **Clear the Date**, and use the **Directory controls** to move to the C:\FILES directory. Set the Search Type to **Search selected directory** and press the **Find Files** button. Use the **All** button to select all 5 files. Then change the current directory to be the source directory that has all 10 files, and press the **Copy From** Action button. It will tell you if any of the 5 files you need are missing from C:\NEWFILES.

Using XCOPY to copy files

With version 3 of WinSince there is a built-in %Copy command for copying files. You can still use the XCOPY command in an Action, as you can any DOS command. With XCOPY it is important to remember the ending \ so that XCOPY does not prompt you on each copy to check if the target is a file or directory. I.e. the Action might look like this: **XCOPY %A G:\%P**

Reclaiming the Original WinSince Actions or File Types

If you somehow lose the Action definitions, the original ones programmed into WinSince can be returned. Edit the Since.ini file and delete the Action lines in the [Since] section. When you start WinSince again the original Actions will be there. If you delete the Button definition lines you will get back the original File Types. If you delete the entire Since.ini then WinSince will regenerate the original distribution Since.ini.

Using the Warning check box to verify Actions and Double Clicks

When first defining an Action, it is useful to set the Warning check box in the Edit screen. This will allow you to see what the final command to be run will be, including the file name. To test the new Action, select one or more files from the top left list box and run your command. If the command to run is not as you expected, you can choose "No" and Edit the Action. Once you are satisfied that your new Action is working, you can reset the Warning check box. For file copying and deleting it is recommended that you always use the Warning option, to prevent file problems. With the Warning on, you will be able to choose "**Yes to All**" on the first Copy or Delete, so that no further warnings are given.

Similarly, you can use the Options menu to set the "Warn before run double click" option. This will show the command to be run when you double click a file name. This is a good way to verify the file associations within Windows.

Using the Warning check box with large Windows programs

If you are going to be running a Windows program on a list of files then you may want to have only one copy of the Windows program running at a time. An example would be running Microsoft Excel on a list of XLS files. By using the Warning box in the Edit screen for that Action, you can complete the first file, then exit the Windows program to find the next file ready to be started. For small Windows programs, such as NOTEPAD, you can have many instances running at a time up to your RAM capacity. If you run out of memory while starting up a Windows program, WinSince will not start the Action on any subsequent files.

An Action Without a File

You can define an Action that does not need any files. The example supplied with WinSince is the running of the Windows Notepad.exe editor without a starting file. When you set up the **Action to Do** leave out any reference to %A, %D, %P, %F, %N, %E, %COPY and %DEL. This will allow the Action to start if no files are selected. If you do select one or more files then the Action will be performed that

many times, but as the Action is defined, there will be no file names passed to the program.

Keeping the Selected Files List

You can use the directory controls to move to another drive or directory without losing the list of selected files. This is one of the integral features of WinSince. Only when you press the **Find Files** button will WinSince get a new list of files. This can let you perform more than one action on the selected files if needed. It also means that you have to be mindful of what files are currently selected: even when you move to another directory, you will still have files from the previous file search selected. This is a unique feature of WinSince.

Keeping the Action Definitions on Screen

If you often enter the Action Edit window to see what the Action definitions are, you can leave it on screen while using WinSince. Clicking on the **Summary** button will hide the edit details, and just show the main Action details. You can continue editing by clicking on the **Detail** button.

Clearing the Date is Important

Using the **Clear Date** button allows you to search for all files of a particular type, regardless of the file creation date. This is very important for many file management tasks, since you will often want to select all or many files in a directory for purposes of deleting or copying them.

Temporary Files created by WinSince

WinSince uses a temporary file, SINCE001.TMP, to store search information. WinSince can have multiple instances running at the same time, so there may be temporary files with other numbers, such as SINCE002.TMP. The location of the file is defined by the environment variable TEMP. WinSince is careful enough to check for low disk quota (900,000), and will notify you if it might run out of free disk space while searching for files. WinSince also creates a DOS Batch File, SINCE001.BAT, in the TEMP area for running DOS batch commands. It also creates SINCE001.IND and SINCE001.SRT for sorting.

Developing using Sheridan Software Systems Widgets

WinSince uses one Sheridan runtime VBX file, SS3D2.VBX. If you are developing with Visual Basic and this Sheridan VBX file, be sure to close WinSince before opening the VB project. If you do not, VB will give you the message "Can't load Custom Control DLL: SS3D2.VBX".

"Sub or Function not defined" error

This error can indicate that the SINCE.DLL or the Visual Basic 3.0 VBRUN300.DLL files are not on the path. For simplicity you can put WinSince in the main Windows directory, which is almost always on the path. This new version of WinSince has a new SINCE.DLL; any copies of the older SINCE.DLL should be deleted.

Checking the Date of a Single File

You can see the file size and creation date by single clicking on any file in the directory control file list, which is just to the left of the directory selecting control. This has no effect on the selected files, and can be useful for comparing details to those in the selected file list.

Refreshing Directory contents after using a DOS action or for the next floppy

If you define a DOS action that affects the number of files in the current directory, you can click on the **"Directory total"** caption above the file list control to refresh the directory file list. This is not necessary when using the built in %Copy and %Del actions. Clicking on the "Directory total" caption is also useful when you are examining a sequence of floppies in the same floppy drive.

Command Line Examples

For command line examples, see the section on **Batch Mode and Command Line Qualifiers**.

Using Command Line Qualifiers

WinSince can run minimized in batch mode, logging files found, performing an action on the files found, and then exiting when done. Because the action to be performed can be any Windows program, DOS command or program, you have complete flexibility in defining what gets done. Its relative date searching lets you search for files created a number of days before the current date. WinSince will find the files you want to work on.

The command line startup instructions allow you to create specialized WinSince instances that **automate your file management**, or **customize your interactive WinSince session**. Once you set up your WinSince Actions, you can create Program Manager WinSince icons to define each of your file tasks.

The 27 start up commands include:

- /Dir=** Directory to start program in
Example: /dir=c:\windows
If /dir is left off, then WinSince will start in directory specified in the Startup Directory specified by the Windows Program Manager property.
- /Filetype=** Type of files to search for
Example: /filetype=*.bat
If /filetype is left off, WinSince uses *.*
Multiple file types can be separated by semicolons: /filetype=*.exe;*.com
- /Subdir** Search down the subdirectory tree from the directory specified in /dir. If /subdir is left off, then only searches startup directory.
- /Drives=** Search the selected drives for the file type.
Example: /drives=cdf would search drives C, D and F.
Non-existent drives are ignored, which can be handy on networks where drives come and go. It will give an error though if no drives are found to be valid.
If both /drives and /subdir are left off, WinSince searches the startup directory.
- /ClearDate** Ignore the creation date when finding files
If /cleardate is left off, WinSince searches for files created today.
- /Since=** Creation date of files to find since. Files created on or after this date will be included.

Absolute date method is handy when you know the date to check.
Note that WinSince now allows an optional time portion with the date:

Example 1: /since=01-30-1993 (find files created since Jan 30 1993 as of midnight, since no time given; i.e. format is **/since=mm-dd-yyyy**)

Example 2: /since=01-30-1993:12:30:00
(find files created since Jan 30 1993 as of 12:30 in the afternoon)

The format when including a time is **/since=mm-dd-yyyy:hh:mm:ss**
Note the colon (:) between the date and the time.

Relative date method is handy for automating searches to run each day:

Example 3: `/since=-5` (find files created since 5 days ago,
as of midnight. **/since=-ddd**)
(i.e. today - 5 days)

The relative date can also specify a number of days and time.
The format when including a time is **/since=-ddd:hh:mm:ss**
Note the colon (:) between the number of days and hours.

Example 4: `/since=-1:12:00:00` (find files created in the last 36 hours)
(i.e. 1 day and 12 hours).

Example 5: `/since=-0:00:01:00` (find files created in the last hour)

/Add

Add files to the File List when searching for files.
The File List will not be cleared when you click on Find Files.
File names are added to the File List if they are not already in
the File List. This control is useful for creating lists of many types
of files or files from many different directory areas. If **/add** is left off
then the File List is cleared when a file search is started.

/Archive

Search for files with the Archive file attribute set.
These are files that have not yet been backed up.
If `/archive` is left off, WinSince ignores the files'
Archive flag setting when searching.

/Search

Search for files when WinSince starts, even if Option is
set to not get files when starting up. Usually you'll want
to include this so that WinSince finds selected files on startup.
You should leave `/search` off if you are using `/dir` and/or `/filetype`
to specify startup values but don't want a file search to delay
the startup of WinSince.

- /NoSearch** Do not search for files when WinSince starts.
This overrides the option to search for files when WinSince starts, and is useful when running Timer events, to prevent files from being searched until the first Timer event occurs.
Note that /NoSearch does not change the value of the Search option, it only overrides it when WinSince starts.
- /Action=** Action number (1 to 20) to perform on the files. The action can be any of the 10 actions defined within the WinSince Edit screen, or the Button Bar Actions.
Example: /action=10 performs action number 10 on any files found.
If /action is present, then all files are automatically selected (just as if you pressed the All button). If /action is left off, WinSince searches for files but does not perform an action on the files found.
- With version 5.0 of WinSince, the first 10 Actions (1 to 10) are the definable Actions. The Button Bar Actions are numbered 11 to 20.
- If /min is present, then the Action performed will run in a minimized window, even if the Action is defined to run full size, etc.
All other aspects of the defined Action are run as defined, so be sure to NOT include any Pauses between or at the end of DOS actions. The reason is that the Action will run minimized, and you will not see the request for keystrokes ("Press return to continue"). The program would sit idle waiting until you maximize the screen and click on OK. You can use the /nowarn qualifier to prevent warnings before the Action starts, but limit its use to proven searches and Actions.
- /Exit** Exit WinSince once the initial file search is done ("batch mode").
This allows you to set up WinSince to perform a single task and then exit. If /exit is left off, WinSince will remain active on the screen when it completes any task you assigned for it ("interactive mode").
- /Min** Run minimized. This is handy for running WinSince as a batch job with /exit when you don't want to see it running.
- /Timer=** Set the amount of time between timer events.
The format is **DD:HH:MM:SS**. For example, to have the timer run every minute and a half use /Timer=00:00:01:30
This is the time between the end of one event and the start of the next event.
- /Timercount=** Set the number of times the timer is to run. Set this value to zero to have the Timer run indefinitely (until press Stop).
If /Timercount is left off, the count defaults to **indefinite**.

- /NoWarn** Suppress the warning defined for any Action run. This is useful for batch runs of WinSince on proven file searches and Actions. **Be very careful when using /NoWarn**: the action warnings protect you from mistakenly deleting or copying files when you don't want to.
- A **yellow caution triangle** is shown on the main screen when this option is active, to remind you that Actions will not be preceded by a Warning.
- /NoCopyWarn** Ignore errors encountered when copying, such as locked files. Serious errors, such as running out of disk space, will still result in a message to the screen. If you are using a log file, all of the warnings not shown can be found there.
- /NoAttribWarn** Ignore errors encountered when trying to set the file attributes of open or locked files.
- /All** Selects all files after a file search, and is equivalent to pressing the **All** button after getting files. You do not need to use this for batch jobs since all files will be selected automatically if you specify an Action on the command line.
- /IfDifferent** Copy the source file only if it is different in size or date than found in the target directory. If a file by that name is not found in the target directory then the source file will be copied. This can be used to reduce the number of files copied. See the Options section also.
- /IfNewer** Copy the source file only if it is newer than found in the target directory. If a file by that name is not found in the target directory then the source file will be copied. See the Options section also.
- /Log=** Name of log file to write files found by WinSince. This can be a file name you specify, or you can let WinSince create a dated log file.
- Example 1: /log=c:\winsince.log (name specified)
- Example 2: /log=c:\%date (create log file by date)
- Let WinSince create a log file according the current date and time. This is very useful for keeping **multiple log files with unique** names, perhaps in a single directory. This is very useful for System Administrators and for auditing. It will add a new log file though for each WinSince session you start, so watch your disk space.
- The format of the log file created is **Wmmddhmm.mss**. For example, for August 4th at 12:34:56 the file is W0804123.456

WinSince will write the type of search and the file details of all files found during the current session. This can be used in batch mode to show what files were found, and in interactive mode to create a list of files, for later printing or analysis. If free disk space falls under 900,000 bytes the log file is automatically closed, so pick a drive with lots of space for the log file: it uses about 80 characters for each file found during the searches.

Note: errors occurring during a search are usually written to the log file, prefaced with ">> ".

/Append Append to an existing log file, as specified in /log qualifier. If the log file does not exist it is created.
My personal preference for interactive file management is to always specify /log=c:\winsince.log on the command line but without the /append. It barely slows things down and is often very handy for checking details about the latest session.

/Find The next file search will include a text search. (* NEW *)
This switch is equivalent to setting the Find Text control on the main WinSince form. It is possible to use the command line to set up /Text= without using /Find in preparation for text searching at a later time.

/Text= Text to find in the files. Note that you can search for one or more words. The text to search for is ended with a } (* NEW *)

Example: /text=find these 4 words}

/NoFindWarn Ignore errors when searching locked or open files (* NEW *)
for text.

/Case The text search will be case sensitive. (* NEW *)
Most text searches do not use this switch, since capitalization usage is often inconsistent.

To set up WinSince to use the command line switches, use the File | Properties menu item in the Program Manager to modify the WinSince command line. I.e. instead of running C:

WINSINCE\WINSINCE.EXE you might change it to show
C:\WINSINCE\WINSINCE.EXE /DIR=C:\ /FILETYPE=*.BAT /SEARCH

You can create multiple instances (icons) for WinSince, each one adapted to do the file searching and actions that you need done. The command line switches can be combined to do a variety of tasks that range from customizing the startup file type to automatically copying files between directories.

In the following examples, the command line qualifiers are spread over two lines for easier reading. In the Program Manager File Properties all qualifiers are on the single line available.

Example 1: You are interested in seeing a list of all TXT files in particular areas:

```
WINSINCE.EXE /DIR=C:\MYWORK /FILETYPE=*.TXT /SUBDIR /CLEARDATE  
/SEARCH
```

This tells WinSince to look down in all subdirectories starting from C:\MYWORK for files with the extension TXT, and to ignore the creation date, i.e. find all .TXT files regardless of creation date. Since no /ACTION is given, WinSince will display the list of files when it appears, leaving it up to you to view the list and possibly choose an Action at that time.

Example 2: You want to review or edit TXT files changed in the last hour, and create a log file showing the list of the files found:

```
WINSINCE.EXE /DIR=C:\MYWORK /FILETYPE=*.TXT /ACTION=3 /MIN /EXIT  
/SEARCH /LOG=C:\MYWORK\CHECK.LOG /SINCE=-0:01:00:00
```

This tells WinSince to look only in the C:\MYWORK directory for files with the extension TXT. The /SINCE=-0:01:00:00 tells WinSince that you are using a **relative** date of one hour in the past. On each of these TXT files, WinSince will run Action number 3 ("Edit"), which in the distribution version of WinSince runs the Windows NOTEPAD editor on the selected file. (You may customize all Actions to do whatever you want if you don't like the defaults). Since /MIN is specified, each Notepad session will appear as an icon, ready for you to work on when ready. Since /EXIT is specified, WinSince itself will never appear to you: it will just start up the Notepad sessions and then exit. The list of files found, along with their dates and sizes, will be put into C:\MYWORK\CHECK.LOG, as requested in the /LOG command. This is an ASCII file suitable for later editing or printing.

Example 3: You want to copy TXT files changed since January 2nd 1993 to a safe directory, and create a log file showing the list of the files found.

This is a similar example to Example 2 except that a specific date has been selected, and the Action is 2 ("Save Copy"):

```
WINSINCE.EXE /DIR=C:\MYWORK /FILETYPE=*.TXT /ACTION=2 /MIN /EXIT  
/SEARCH /LOG=C:\MYWORK\CHECK.LOG /SINCE=01-02-1993
```

This tells WinSince to look only in the C:\MYWORK directory for files with the extension TXT that have been created since January 2nd 1993. The /SINCE=01-02-1993 tells WinSince that you are using an **absolute** date. Action 2 is defined in the distribution kit as %COPY %A %D:\SAVE\%P\%F . If we use C:\MYWORK\HELLO.TXT as one of the files found, it would be copied to C:\SAVE\MYWORK\HELLO.TXT . The subdirectory structure is retained.

Setting up multiple icons for WinSince on the Windows desktop

As you come up with your own list of file tasks that you would like to automate, you can add copies of the WinSince icon within its Windows group. To do this, hold down the Ctrl key while you click and drag the WinSince icon to a free spot within its group. Modify its File Properties with the command line qualifiers, and change its title to show the meaning of the task it does.

Warning: You have complete flexibility in creating and running your Actions. This also means that you must exercise care in using them. Testing your Actions in interactive mode with the Action's Warning option turned on is a good way to be sure that /Action and other command line qualifiers do what you want on the right files. Be mindful of how many files may be found in a search or passed to an Action: you could run out of memory or disk space. Be careful of which files are currently selected, since they are from the last file search and not necessarily from the current directory.

Copying and Deleting Files

WinSince provides built in file copying and deleting commands: **%Copy** and **%Del**. Both commands are used within Actions defined in the Edit screen.

Use the "Warn before Starting" Action Warning:

For both %Copy and %Del actions, it is recommended that you use the Warning option in the Edit screen so that you are sure that you are performing the action you really wish. It is recommended that you **not** use the "**Suppress warning before Action (!)**" or **/NoWarn** options, except on proven file activities. With the Warning in place, you will get a notice of what action is about to occur, and be able to skip the action for this file, cancel the operation completely, or to accept the operation for the current file or all further selected files.

%Copy

The format for %Copy in an Action is:

%Copy From_File To_File where From_File and To_File have one or more file name substitution tokens to represent the files being copied (%A, %D, %P, %F, %N, %E).

Example: To copy all selected files to the root directory on the A: drive:

%Copy %A A:\%F where %A represents the entire path and file name, and %F the File name.

When passed the file C:\WINDOWS\SYSTEM\README.TXT the action performed will be:

%Copy C:\WINDOWS\SYSTEM\README.TXT A:\README.TXT

As long as the Action Warning is set, you will see which file is going to be copied where. You will also be shown if the source ("from") file is newer or older than the target ("to") file, and the file sizes if available. Very often when copying you want the source file to be the newer file.

For a variety of other examples of the %Copy command, see the section regarding the **Actions in the Distribution Kit**.

%Del

The format for %Del in an Action is:

%Del Filename where Filename will usually have one or more file name substitution tokens to represent the files being copied.

Example: To delete all selected files:

%Del %A where %A represents the entire path and file name.

When passed the file C:\WINDOWS\SYSTEM\README.TXT the action performed will be:

%Del C:\WINDOWS\SYSTEM\README.TXT

As long as the Action Warning is set, you will be asked if you wish to delete the current file.

Actions in the Distribution Kit

WinSince provides 20 Action definitions. 10 of these Actions can be redefined by you, to suit your own file management requirements, and 10 are Button Bar Actions. The following is an explanation of the uses and possible alternate ideas for these Actions. There have been no changes in the supplied definable Actions from version 4.5 to 5.0, although you may have customized your own Actions.

Definable Actions (can be changed):

<u>Button Caption</u>	<u>Action Definition</u>
1) Copy to A:	%Copy %A A:\%F Copy the selected files to the root directory of the A: drive. It does not use the %P Path token so all files selected will be copied into the single root directory, overwriting any previous file that had the same name. I recommend using the Action Warning for all %Copy uses.
2) Save Copy	%Copy %A %D:\SAVE\%P\%F Copy the selected files to the same drive, but copy them below a subdirectory called SAVE. For example, the file C:\WINDOWS\README.TXT will end up copied to C:\SAVE\WINDOWS\README.TXT. This command is useful for making a copy of important files. You can easily Edit the Action to change the SAVE directory to another directory path, including a longer one, such as %Copy %A %D:\THIS\THAT\%P\%F
3) Edit	Notepad.exe %A Pass the file name to the Windows text editor. This is an example of calling a Windows program from WinSince. The Notepad editor is a handy one for small files (under 64K). For larger files you may wish to use the Windows WRITE.EXE: WRITE.EXE %A
4) Del	%Del %A Delete the selected files. Be sure to set the Warning for file deletions so that you get a warning before the deletion occurs. Although you can dream up some interesting variations on the %Del command, I think that this is the version most people will use.

Button Caption

Action Definition

5) Copy To

%Copy %A %CurDrv:\%CurDir%\%F

Copy the selected files to the drive and directory pointed to by the WinSince directory controls. This is a lot like drag and drop except that it is point and click. This is very useful for copying files from one place to another without having to redefine the Action, as you have to in the Save Copy Action. Note that all files go into the single target directory. This can generate an error if some of the files in your File List are from the directory you are copying to, since you cannot copy a file onto itself.

6) Attrib +A

%Attrib %A +A

Set the file attributes so that the Archive bit is set for the selected files. The Archive bit is used to tell programs such as your backup program that this file has changed or needs to be backed up. Some people will want to change this Action to be **%Attrib %A -A** to use for resetting the Archive bit after they copy the file to a safe location, in lieu of doing a backup. As described in the section on using %Attrib, **any setting of file attributes can have serious effects**, so be sure about what file attributes you are updating. As a general precaution for all PC users, I recommend fully backing up your hard drives on a regular (weekly) basis.

7) Dir

Dir %A

It doesn't get much simpler than this. The Action is defined to use a DOS batch file, with a PAUSE between each Dir. When you run it on a list of selected files it opens a DOS Window and runs the DOS Dir command on each file. **This is a good action to replace with your own.**

8) Edit New

Notepad.exe

This is an example of running a Windows program without a file name. If no files are selected WinSince still starts the program. You could redefine the Action to run Excel.exe or some other program. If you do select more than one file then WinSince will start that many copies of the program but it won't pass the file name because none is specified in the Action.

Button Caption**Action Definition****9) Copy Down****%Copy %A %CurDrv:\%CurDir%\%P%\%F**

This is a variation on a subdirectory tree copy but with a difference. Its use is mainly to copy a set of files to another drive while maintaining the same directory structure. For example, you could find all INI files on the C: drive, then point to the D:\ root to place a copy of them on D: with the same directory structure.

Because the files you find could be any distance down from the root directory, WinSince appends the file path to the current directory. For this reason, you cannot perform this copy to the root directory of the same drive, because in effect this is trying to copy the files on top of themselves. As an example of what not to do, if you have C:\WINDOWS\SYSTEM\README.TXT selected and you are pointing to the C:\DOS directory for the Copy Down Action, you will create the file C:\DOS\WINDOWS\SYSTEM\README.TXT.

10) Action 2+6**%Action2+%Action6**

This is an example of a combined Action. It is equivalent to running Action 2 (Save Copy) on the selected files, followed by Action 6 (Attrib +A). It illustrates a new capability in version 4.4 of WinSince: to create a chain of Actions. This is useful for times when a single Action definition does not do everything you want to the selected files. See the help section on Editing the Actions for further information on using %Action.

In version 4.4, Action number 6 ("Attrib +A") replaced the "Have File" Action, and Action number 10 ("Action 2+6") replaced the "Copy From" Action from previous WinSince versions:

<u>Button Caption</u>	<u>Action Definition</u>
Have File	If exist %CurDrv:\%CurDir%\%F echo %F exists in %CurDrv:\%CurDir This command is a DOS Action example that will show you if a version of the selected file can be found in the directory currently pointed to. The file may be of different size and date. If you are proficient at DOS batch file creation, you could create all sorts of batch files that you pass the selected file to, such as MYTHING.BAT %A which could use the DOS %1 parameter to do something with the file.
Copy From	%Copy %CurDrv:\%CurDir%\%F %A This Action is described below, in the Button Bar section.

Button Bar Actions (cannot be changed, but can be combined in definable Actions)

Note: All Button Bar Actions have the **Warn before start** flag set in their definitions.

11) Copy to A:	%Copy %A A:\%F This is the same as Action 1 defined above.
12) Copy To	%Copy %A %CurDrv:\%CurDir%\%F This is the same as Action 5 defined above.
13) Copy Down	%Copy %A %CurDrv:\%CurDir%\%P%\%F This is the same as Action 9 defined above.
14) Copy From	%Copy %CurDrv:\%CurDir%\%F %A This is the reverse of the Copy To Action and is very handy for refreshing the list of selected files from the directory pointed to. For example, a network administrator could select a given client's files, then point to the main Network directory to Copy From. In other words, it replaces the original selected files with those in the directory pointed to. Because you may not find copies of all of the selected files in the source directory, WinSince will tell you the total number of files that it was unable to copy. Sometimes it will be acceptable that not all files were found, such as when you are looking for new versions of files that MAY have been created.
15) Edit	Notepad.exe %A This is the same as Action 3 defined above.
16) Edit New	Notepad.exe Start the Notepad editor without a file.

17) Write

Write.exe %A

Pass the file name to the Windows WRITE editor. WRITE is a very usable editor, and useful for handling files larger than 64K, which the Notepad editor can not handle. If you send WRITE the name of a text file, it will ask you if you want to convert it to Write (.WRI) format. Usually you do not, and should choose "No Conversion".

18) Attrib -A

%Attrib %A -A

Reset the file attributes so that the Archive bit is **not** set for the selected files. This is used to tell programs such as your backup program that this file has **not** changed and does **not** need to be backed up. This is opposite to Action 6 (**%Attrib %A +A**). As described in the section on using %Attrib, **any setting of file attributes can have serious effects**, so be sure about what file attributes you are updating. You might use this Action after copying a file to a safe location, such as onto a floppy that you will be keeping.

19) Attrib +R

%Attrib %A +R

Set the file attributes so that the file cannot be changed or deleted. As described in the section on using %Attrib, **any setting of file attributes can have serious effects**, so be sure about what file attributes you are updating. You might use this Action on a file that should never be changed, such as a stable list of names. If you ever do need to change or delete the file, define an Action as **%Attrib %A -R** to reset the read-only flag.

20) Del

%Del %A

This is the same as Action 4 defined above.

Drag and Drop Capabilities

WinSince provides drag and drop methods for running an Action on a single file, adding a file to the File List, and setting the Find Since date from a file.

Run an Action on a Single File

You can drag and drop a file from the directory control onto an Action button. This is very handy for running an Action on a single file, since there is no need to go through the process of using the Find File button and selecting the file from the File List. The same Action warnings are in effect.

Add a File to the File List

You can drag and drop a file from the directory control into the File List directly, without going through the process of using the Find File button. This makes it easy to create a diverse list of files quickly. (You can also use the Add Files feature to create diverse lists). Dropped files are added to the beginning of the File List when not in paging mode, so it is easy to see what has been added. In paging mode, dropped files are added to the end of the File List for greater speed. The total size of the File List is also updated.

Set the Find Since Date from a File

You can drag and drop a file from the directory control onto the Find Since date. This sets the file search date to that of the file dropped, and is useful for checking for files created since that particular file was created.

Setting File Attributes with %ATTRIB

WinSince can be used to set the following attributes on a file:

<u>Attribute</u>	<u>Meaning when attribute is Set</u>
Archive	File has been changed and not backed up
Readonly	File can be read but not changed or deleted
Hidden	File is not visible in directory listings (use with care)
System	File is for system use, and is not visible (use with care)

The %ATTRIB command can be used in an Action to set or reset these file attributes on files in the File List or on a single file dropped from the Directory List onto an Action containing %ATTRIB.

The format for using %ATTRIB is:

%ATTRIB FileName +/-Attributes

Example: %ATTRIB %A +A-R
For each file selected, set the Archive attribute and reset the Readonly attribute.

If the file already has the attributes described in the %ATTRIB command then no change is made to the file, and no Action is performed or warned about before being done.

Once completed, the File List entry for the file is updated to show the new file attributes.

Ignore Attribute Setting Errors Option

If a file is locked or opened by another program then it is not possible to change its file attributes. WinSince will notify you of this and give you the option to end processing of further files. If you know that certain files will be locked or open and wish to not see the warning then use the Ignore Attribute Setting Errors option in the Option menu. From the command line you can include /NoAttribWarn when you start WinSince.

All file attributes are significant, and should be changed only when sure of the effects:

Resetting (removing) the **Archive** file attribute would mean that the file would not be detected as changed during an incremental backup of your files. You may still want to reset the Archive attribute (-A) if you know that a file has been copied to a safe location. WinSince provides the option to search for files that have the Archive flag set.

Setting the **Readonly** file attribute will prevent you from saving an updated version of a file, such as from a word processor or spreadsheet. The Readonly flag is a useful setting for files that do not change and that you don't want people to change.

Setting the **Hidden** or **System** file attributes will hide the file from view from WinSince and other programs, and is generally reserved for use by system-level programs such as the operating system. **I recommend not changing the Hidden or System file attributes.**

As a general precaution for all PC users, I recommend fully backing up your hard drives on a regular (weekly) basis.

Using the Timer

WinSince 4 introduces the Timer, which will let you trigger a File Search and Action at regular intervals. For example, you might want to keep a background copy of WinSince running once an hour, deleting .BAK files or copying new files to a safe area.

You can also use the Timer without an Action. For example, you could audit activity on your system by using the log file to track which files have been created in the last hour.

Start or Stop Timer

This menu item is used to toggle between the Timer being **on** and **off**. Before starting the Timer the first time, you will need to set up the interval between Timer events using the second menu item, Set up Timer and Action.

When the Timer is running, you will see the word **Running** beside the Timer menu heading on the main WinSince screen. The **Timer History** form will also be displayed, showing a summary of the Timer information (described in a later section).

You can **stop the Timer** by using the same Start or Stop Timer menu item, or by clicking on the **Stop** button on the Timer History form.

While the Timer is running, the **controls** on the WinSince form **are disabled**, since they are under the control of the Timer. If you need to do some interactive work while the Timer is running, you can start another copy of WinSince. One of the strengths of WinSince is the ability to have multiple icons set up for various purposes. One of these icons should be set up for regular, interactive work.

The Timer copy of WinSince can be **minimized** while it runs (you can use the /min command line switch or click on the minimize buttons).

Set up Timer and Action

This is Timer menu item used to tell WinSince how often you want the Timer to go off, how many times to run the Timer, and what (if any) Action to do on the files found.

Time between events: Set the **time between the end of one event and the start of the next event** using the spin controls or by typing in the fields. The Days field allows any number of days. Hours can be 0 to 23, and Minutes and Seconds can be 0 to 59. You can quickly reset an individual time component to zero by double clicking on the field. From the command line use **/Timer=DD:HH:MM:SS**

While WinSince is searching for files and performing an Action on the files found, the Timer is not marking time. Most file searches of single directories take only a few seconds. When the event is complete, the Timer once again starts monitoring the selected time between events.

Action to run on files: This can be a value from 0 to 20. The Action to run is displayed beside the Action number, though longer Actions may be truncated in the display. From

the command line use **/Action=number**

Remember that Actions often have a Warning before they start. You will often want to set the Suppress Warning before Action option before starting the Timer (or **/nowarn** on command line).

Number of times to run: If you leave this value at zero then the Timer will keep running until you stop it. This can be set from the command line with **/TimerCount=number**

OK button: Accept the new Timer values. These values will be used when the Timer is started.

Cancel button: Ignore any changes made to the Timer values. This is useful if you accidentally zero the wrong values. You can then go into the Set up Timer and Action menu item again to work on the original values.

Zero All: Set all Timer values to zero. If you leave the values all zero the Timer will not start.

Timer History

This menu item pops up the Timer History form, and is the same form displayed when you start the Timer. It summarizes the Timer values and shows the ending values from the previous Timer session, if there was one.

Timer Interval: The time between the end of one event and the start of the next event. This is in the format **DD:HH:MM:SS**. For example, 00:01:10:00 would be one hour and ten minutes between events.

Action to Do: This will either say **No Action** or the number and description of the Action to do after the File Search. For long Actions, the description may be truncated.

Start Time: Each time you start the Timer that time is shown here.

History: The **History List** shows the times that the Timer events occurred, with the most recent event at the top. Up to 100 event times are maintained. If you are using a log file, all of the times can be found there also. This same History can be seen by using the **Timer History** menu item when the Timer is not running.

Times Run: This is the number of Timer events that have completed since the Timer was started. A Timer event is considered to complete when both the File Search and Action (if any) are done. It is reset to zero each time the Timer is started. When the **Times Run** value equals the **To Run** value then the Timer session will end. If **/Exit** was given on the command line then WinSince will stop, otherwise it will return to interactive mode. If you specified **/Exit** and continuous

Timer events, then stopping the Timer leaves WinSince on the screen in interactive mode.

OK button:

This makes the Timer History form disappear but does not affect the Timer if it is running. If you are running the Timer you may wish to minimize the main WinSince form and keep the Timer History visible, or also minimized for checking. You can show the Timer History form again by clicking on the Timer History menu item.

Stop button:

This will stop the Timer. It functions the same as using the **Start or Stop Timer** menu item. It will also remove the Timer History form from the screen.

Example of using the Timer from the command line

Suppose you want to look for all new files created in the C:\WORK directory every hour, and copy them to a safe directory. The command line for the WinSince icon could be set up as follows:

```
WinSince.exe /filetype=*. * /dir=C:\WORK /since=-0:01:02:00 /timer=0:01:00:00 /ifdifferent  
/nowarn /action=2
```

Note how the /since switch is set to look for files created in the last 62 minutes rather than 60 minutes, but the /timer switch is set to 1 hour (60 minutes). This is for two reasons: the /timer value is the time between the end of the previous timer event and the start of the next, and the event of searching for files and copying them may take a minute. Also, under Windows, a timer may not get a chance to be activated if another program is not releasing the CPU. The /ifdifferent switch offsets this by not copying a file of the same size and date previous copied. You could even set /since to be a longer time, such as /since=-0:02:00:00 (last 2 hours) to be sure. The /nowarn switch prevents the Timer from being interrupted by confirmation warning messages.

Use a unique log file for each Timer instance

When using the Timer, it is very handy to create a log file to keep an audit trail of what was done. Very often the Timer is used minimized in the background, so a log file is a good source of information about what files WinSince found, if any errors occurred, and at what time.

Each WinSince Timer icon you set up can have a unique name to distinguish it from the usual interactive log file (often set to /log=c:\winsince.log). For example, you might use /log=c:\wstimer1.log or /log=c:\wstimer2.log. This will prevent an error should a Timer version of WinSince try to open the locked log file, and to provide the audit trail of what happened.

As a reminder, you can use /log=c:\%date to have WinSince generate a unique log file name, though you may want to create these in a separate subdirectory, such as /log=c:\log\%date.

Searching for Text

WinSince is able to **search the selected file type for a text string**. For example, you might want to search all *.TXT files for the phrase "to my friend". Searching for text in WinSince is an added constraint when searching for files to add to the list. If you have selected a Since Date of yesterday, a file type of *.TXT, and to search drives C and D, then finding "to my friend" in those files is simply an additional check that WinSince does before adding a file to the File List.

WinSince considers the text search a success if it finds **any occurrences** of the text string in a file.

To set the text to find, click on the **Search** menu, and then the **Text to Find** menu item. The form provided has a place to enter the text to find, as well as some options.

The form duplicates the Search menu options to use a **case sensitive search** (which is not common), to **ignore warnings** if WinSince tries to search a locked file, and to **find text the next time** a file search is done.

You can control including the text search from either the Search menu ("**Find text on next search for files**") or directly from the WinSince main form by setting or resetting the **Find Text** option control.

Once a text search has completed, the search statistics can be seen in the Search menu's **Search History**. This shows what text was searched for, how many files were found and how many were searched. It also shows which files could not be searched due to being locked or open.

Paging Mode and VCR Controls

When there are more than 650 files in the File List, WinSince uses a paging technique to manage the list. A **yellow page** in the bottom right corner of the File List indicates that WinSince is in Paging Mode. As you move through the File List, an hourglass cursor will indicate when WinSince is paging additional records into the list. The upper 450 files in the File List are not paged, so paging occurs most as you approach the bottom of the list. Sets of 50 records are loaded at a time, so you will see the loading occur every 5 screen lengths that you page up or down. When there are fewer than 650 files, all files can be shown in the File List and paging mode is not needed. WinSince can handle up to 7000 files; enough for an entire hard drive.

VCR Control

The main difference you will find with paging mode is that the vertical scroll bar does not show the proportional position within the list, although it operates normally. To assist movement through the File List, a set of four **VCR controls** has been added. These controls can be used whether WinSince is in paging mode or not.

From left to right, the button functions are:

Go to the top of the File List

Go up one page (same as clicking on the upper part of scroll bar itself)

Go down one page (same as clicking on the lower part of scroll bar itself)

Go to the bottom of the File List

To move up or down by a single record, use the usual vertical scrollbar arrows, located at the top and bottom of the File List scrollbar.

Position in the File List

When you **click on the yellow page**, a message box will tell you the line number of the file currently at the top of the File List. It will also show you the percentage through the File List that the uppermost file represents. For example, when at the very top of the list of files, the position will be 1 and the percentage through the list is 0. When at the very bottom of a list of 850 files, the position will be 842 and the percentage will be 99.

Selecting Files in Paging Mode

Use the **Control** key ("**Ctrl**") in conjunction with the mouse to select files in paging mode. If you click on a file without the Control key down, then all other files become unselected. You can select additional files by using the VCR controls to get to other parts of the File List, and then use the Control key plus mouse again.

The **All** and **None** buttons above the File List are also useful for selecting all of the files, and unselecting all files.