The Click! Filer Manual- Version 1.1

MODIFICATIONS: version 1.1B (11- 5- 90) see end of document appendix 1.1B : changes and bug deaths

Introduction

Welcome to Click! Filer version 1.1 for Microsoft Windows 3.0. This program is designed to provide the combined features of Program Manager and File Manager in one convenient user interface. Click! Filer is brought to you by TrogloByte and is Freeware. Click! Filer calls other Shareware programs which provide archival utility, if you have these programs I urge you to please register and support them as they are fine examples of programmer ingenuity. My thanks to all the Shareware and Freeware Programmers out there providing useful software to the PC community.

This version is much changed from the original in appearance, features and is considerably more bug free. Many users have provided helpful bug reports and suggestions, they are too numerous to mention, so I'll send a group Thank You! to all who have contributed their input to this program. Hopefully you will find this version easier to use and understand. Please feel free to distribute Click! Filer on Networks and Bulletin boards. All that is requested is to please keep this ZIP file intact and charge no fees for accessing it other than reasonable transmission costs or mailing fees. Now on with the documentation...

User Interface

The disk system in your PC is presented as 4 list boxes which are grouped as Source and Destination. Each group contains a File List and a Directory List. Double clicking on an entry in a Directory list box will take you to that directory. Files may be selected individually or in multiples with single clicks. Clicking on a highlighted entry in a listbox de-selects that entry.

Also in the Click! Filer window are editboxes, pushbuttons, and arrow-selectors. Editboxes allow entry of text, pressing TAB after typing enters the text into the appropriate variable. The arrow-selectors pan through selections such as panels of custom buttons. Pushbuttons are as you expect, click them and something happens!

Using the Source Directory controls

Under the Word SOURCE is the current Source directory. Actions such

as Copy and Move act on the items highlighted in the Source listboxes. Beneath the source directory path is an editbox. This editbox provides several functions.

FUNCTION 1: Change current Source Directory

Enter a new path and press TAB, Click! will update the Source Directory listbox and the Source File listbox to reflect the requested change. The current wildcards will remain in effect.

FUNCTION 2: Change current Source Directory and Wildcard

Enter a new path and wildcare and press TAB. For example C:\newdir*.txt [TAB] will change to the directory C:\newdir and display all files ending in .txt

FUNCTION 3: Change Wildcard

enter a new wildcard string and press TAB. The current Source directory will remain the same but the Directory and File lists will be updated. Only the file matching you new wildcard will be displayed in the Source File listbox.

Directly beneath the editbox are the two Source listboxes. The one on the left is the File listbox and the one immediately to it's right is the Directory listbox.

Source Directory Listbox

This box displays the available drives, subdirectories. If the current directory is not the Root Directory of it's respective drive, a parent selector [..] is provided to move up the directory chain. Double clicking on an entry in this box will change the current directory as requested. Some Click! Filer functions can act on a Source Directory, For such functions you can single click on a directory listbox entry and highlight it. Clicking again on that entry de-selects that entry.

Source File Listbox

This box displays the files which reside in the current Source directory. If the current Source Wildcard or Mask is not *.* then the files listed are only those which match the current Source Wildcard. Single clicking on a file, highlights that file. You may highlight more than one file. Clicking on a highlighted file de-selects that file. Many Click! Filer

functions act on the highlighted Source Files. These functions will be described in subsequent parts of this manual.

Source Directory Command PushButtons

Directly beneath the Source Listboxes are 8 push buttons. These buttons are labeled (All, None, Ren, Del, Edit, Run, MkDir and RmDir). Each button acts upon elements highlighted in the Source Listboxes.

Source All Pushbutton

This button highlights all files displayed in the Source File Listbox.

Source None PushButton

This button de-selects all files in the Source File Listbox.

Source Ren PushButton

This button allows you to rename the highlighted file in the Source File Listbox, or the first highlighted file in the Source File Listbox. Upon pressing this button a dialog is presented which displays the name of the highlighted file, a new name editbox and two buttons (Rename and Cancel). Enter the new name you wish for the file and press TAB. To execute the Renaming of this file press the Rename Button. If you wish to abort the process, press the Cancel Button.

Source Del Pushbutton

This function deletes all highlighted files in the Source File Listbox.

Source Run PushButton

This button will attempt to run the file which is highlighted in the Source File Listbox. If more than one file is highlighted it will attempt to run the first file which is highlighted. You may run Windows 3.0 programs, DOS programs and BAT files. Wherever a .PIF file exists for a DOS program it will be used if it resides in the DOS path of your machine. Otherwise the Windows Default.PIF is used. Such programs will return to a standard DOS style screen and execute. Upon completion they will return to Windows and Click! Filer. There are ways to optimize programs for use under Windows and such techniques appear later in this document.

Source Edit Pushbutton

This button will launch the editor of your choice and pass it the files highlighted in the Source File Listbox. Your editor must be able to accept command line arguments. Due to restrictions within Windows, command lines of only 128 characters can be passed to a program. If you routinely edit more files than will fit in this argument, use of the custom buttons and wildcard arguments may provide you with a convenient method of loading all the files you require. This option will be discussed later in the manual. The editor which is called by Click! Filer is user definable. The process is described in he Defaults & INI section of this manual.

Source MkDir PushButton

This button will provide a dialog for creating directories. The Make Directory Dialog provides a Directory Listbox and an editbox for naming new directories. Three buttons are also in the dialog (Make, OK, Cancel). Double click in the Directory Listbox to maneuver through the disk drive directories. The current directory will initially be the Source Directory. One you have moved to the desired directory, enter the new directory name in the edit box and press TAB. Click on the Make pushbutton to create the new directory. You can continue making directories as long as you wish. To exit press OK or Cancel.

Source RmDir PushButton

This function has two modes of operation. If a directory is highlighted in the Source Directory Listbox, Click will attempt to remove that directory. If successful, the directory will be removed from the listing in the Source Directory listbox. If no directory is highlighted in the Source Directory Listbox, a dialog will appear. Within this dialog you can move through directories in a listbox by Double Clicking on entries. Single clicking on an entry places it's name in an editbox. Once a name is in the editbox you can click on the Remove pushbutton and click will attempt to remove that directory. You can remove as many directories as you wish and exit via the OK or Cancel button. You can also enter directories in the editbox via the keyboard.

Other Predefined PushButtons

Beneath the previously discussed buttons is a row of 7 buttons. The buttons on the left and right ends of the row change their labels to reflect the current Wildcards. The left button is the Source WildCard, while the right button is the Destination WildCard. Pressing these buttons executes a rescan of the respective directory. Sometimes the directories and files listed in a listbox may not update as you expect. This provides an easy way for you to rescan and be assured of a

current listing. Each time you enter new wildcards in the editboxes for Source or Destination, the wildcard buttons are updated. An example use is when you change floppies in the A: drive. This button will rescan the new disk and display the current directory listing to the wildcard contained in the button text.

The second button in this row is the Copy pushbutton. This button will attempt to copy the highlighted files in the Source File Listbox to the current Destination Directory. If no files are highlighted and a Directory is highlighted in the Source Directory Listbox, a Copy Directory Dialog will appear. This dialog lists the current destination directory and the source directory highlighted in the Source Directory Listbox. As the process unfold, the status is reflected in text at the bottom of the dialog. There are 3 options presented as check boxes in the dialog. These interact with each other to accomplish different forms of copy.

OPTION 1: Make selected directory in dest
OPTION 2: Copy all subdirectory files
OPTION 3: Create subdirectories in destination

EXAMPLES :

1 2 3

X X X A directory of the same name as the highlighted source directory will be created in the destination directory. Each of it's subdirectories (and their subdirectories etc..) will be created to match the source. All files from each directory will be copied o it's counterpart in the destination.

X X All the files in the highlighted source directory will be copied to the current destination directory rather than a similarly name subdirectory in the destination. All source subdirectories will then be created and duplicated as above.

X The highlighted source directory will be created under the same name in the Destination directory and all it's files copied to it's counterpart inthe destination. No subdirectories or their counterparts will be copied or created in the destination.

X X The highlighted source directory will be created in the destination directory and all of it's files and all the files in it's subdirectories will be copied to the newly created directory in the destination. Effectively the files are collected into only one directory.

The Sixth button is labeled Move. It's actions are similar to the Copy button. The essential difference being that the sources are deleted after copying the files (and or directories).

The 3 buttons in the middle of the row are Source/Dest swap controls. The button label '>' will set the current source directory into the destination. Likewise the button labeled '<' will set the Destination directory into the source directory. The middle button labeled '<>' swaps the source and destination directories, allowing you to easily toggle between two directories.

Custom PushButtons

The last 3 rows of buttons in the Click! FIler Window are User definable buttons. To the right of these buttons is an arrow-selector which will move through the defined panels of buttons. The initial setup of Click! Filer creates 2 panels of 12 buttons for you to use. You may define up to 99 such panels. each panel can be named and the name of the current panel showing in the window is listed in the Click! Filer TitleBar. Clicking on a defined button will execute that button's function. In addition each button can have a hotkey defined, this hotkey will be detected by Click! Filer even if in icon form and executed as per your definition. These definitions can include Windows 3.0, DOS or BAT files. The method of configuring these buttons is described in the Defaults and INI section of this Manual

Byte Size Displays

Directly above the Source Listboxes is a text display of the size of the currently selected files. This may be configured to display as Bytes, KiloBytes or MegaBytes. You can change this via menu options in the Click! Filer menu bar. Directly above the Destination Listboxes is a text display of the available free space in the drive currently referenced by the destination path. It can be similarly configured via menu options.

Defaults & INI files

Click! Filer must have a file called Click.INI in the Windows directory of

your computer. When you first run Click! Filer, it will check this directory to find this file. If this file does not exist you will be presented with a sequence of dialogs which describe the process of building the INI file. While building the panel ini entries, a bit of time is required. Do not panick. After creating the panels entry, Click! duplicates it in a binary form for quick access bythe program. This file will also reside in the Windows directory, do not delete it as Click will have to recreate it. Do not edit you Click.ini file in an editor. Due to responses by Windows when accessing this file, it is possible that errors can occur when using it. The result is unpredictable and could result in loss of your group files. To allow easy editing of the INI file, Click! Filer has many places where a SETINI button exists. Using this button places the new definitions into the file and assures their correct format.

Click Defaults

The principle Click! Filer defaults are:

Initial Source Directory This will be the Source Directory whenever you start the program

Initial Dest Directory This will be the Destination Directory whenever you start the program

Initial Wildcards for both Destination and Source

Editor You choice of editor to respond to the edit button in Click! Filer

You can configure these options by selecting from the Defaults Menu the option ' Edit Click! ' . A dialog will appear providing editboxes to enter the values. Once completed you must click on SET INI to write these to the click.ini file. OK will not do this for you.

At any point in the program you can select the ' Load Click ' option from the Defaults Menu to re-read the defaults fromthe click.ini file and set them into Click! Filer.

Custom Defaults

Selecting Custom Defaults from the Default Menu presents the Custom Buttom definition dialog. This dialog while somewhat imposing is relatively simple to use. The first component is a button selection group. This is 12 radio buttons each representing the 12 custom buttons at the bottom of the Click! Filer Window. Select the desired

button by clicking on it. After clicking on a button, it's associated properties are displayed in the rest of the dialog. One dialog editbox allows entry of the Label you wish for the button. Enter the text and press TAB. One editbox allows you to enter the program name to execute. If not in the path, enter the full pathname for this program. You can run with any of 4 argument forms. Four radio buttons are grouped together. The simplest is to have no arguments. You can run with source arguments, in which case the higlighted source files will be passed to your program in the command line. You can run with dialog arguments, this option will offer a dialog and edit box to allow you to enter the desired arguments. The last option if to run with fixed arguments. If you use this option, enter the fixed arguments into the editbox provided. If no fixed args are need then leave this entry 'none'. Do not leave this edit box empty. Click! Filer should protect itself from this but in the past had problems with empty INI fields. Finally is the assignment of an associated Hotkey combo. If you wish to have a hotkey, check the 'Use Hotkey' check box. Click on the SetHotkey button and a dialog will be presented. This dialog allows you to press the keys desired. Once they are correct select OK. and you are configured. ALMOST! Before going any further click on SET INI to place this button's program in the INI file. If you click on a new button radio before doing this, all will be lost. Repeat this procedure for each button in turn.

As stated before you can have multiple panels of buttons defined. You can pan through the panels via the Arrow selector near the top of the dialog. If you need to add more panels use the lower Arrow selector to add panels. The pan will then be able to go to the new panels. When a new panel is selected, Click! Filer will pause and create a template for this panel in the INI file, please wait. To name a panel enter it's name in the editbox directly below the panel selector. When completed, select OK.

Archival Support

Click! Filer supports use of several archival utilities which are shareware for the PC. Please support these programs if you use them. Info to contact their authors is available in the ABOUT dialog associated with each function. Currently supported are PKZip, PKPak, Pak, Arc and Lharc. Each of these can be called via the Options in the Archive Menu of Click! Filer. When archiving, Click! always acts on the current Source Directory and places the archive in the current Destination directory. Have these correct before invoking the archival utility. Supplied with this Zip are suggested .PIF files for each of the utilities. Before using you must modify these to match the correct path for that file. Windows can then provide a window for each to operate.

This will improve the look and feel of operation although this is not required as long as the programs are in your DOS path.

The features of each archiving and unarchiviing dialog are best understood by reading the docs for the associated program. The terms used for dialog elements in Click! Filer match those of the archive's author.

The final Menu Options

About You guessed it, about us!
Exit Obvious

Optimizing Custom Programs for Use in Click! Filer

Not all DOS programs actually run under Windows 3.0, although many will. Experiment with each. Use the PIF files provided as a template for programs which you want to run. The PIFs are especially good for small utilities, other programs will require a more definite PIF and extended feature definition. Consult your Windows Manual for help with the PIF Editor.

Batch files will work with Click, I use them all the time. But they work best when written to guarantee success. It is best for BAT files which act on known files in known directories to always start with the following

EXAMPLE:

```
c:
cd c:\mydir
```

The above guarantees that the current drive is the desired drive, likewise changing the the desired subdir (in this case c:\mydir) will be advised. The actual current directory can be difficult to gauge. If the last thing you did was change the destination, the system may be CD'd to that directory. The above technique works well even when setting up to compile programs etc...

Whenever many files of a similar nature are require to be passed to the program, use the Dialog arguments form to take advantage of Wildcard entries or passing listfiles to the program.

Conclusion

I hope you enjoy using Click! Filer. Your comments and suggestions are welcome. Address them to TrogloByte on GENIE network. More features are sure to be added as We explore Windows 3.0. The complexities of directly addressing the Operating system and hardware preclude some of the more common functions such as Disk Optimization and would likely be too dangerous to attempt under Windows. Text has been addressed via the editor and as you probably have your own favorite I feel adding code size to Click! doesn't help anyone. Sound and graphics are areas which you may be interested. Your needs will never be known unless you communicate them, so get on the network and let us know what you want.

APPENDIX: 1.1B

1. When running under 1024x768 mode users got a click dialog with the right edge and the bottom truncated. This no longer plagues us. This bug is squashed. If you other folks have display problems. The CGA display is just not gonna happen, not enough room

2. Users of the Windows Development Kit, experienced crashes when running Debug Windows. This disappointing state of affairs is ended.

3. Source bytes had an annoying habit of ending in .512 all the time. Repaired the problem.

4. Recompiled to optimize on the 80286 microprocessor. Since Windows really needs at least a 286 to run, I hope this adds some performance to the program. If not, well we tried.