CT-Shell for Windows

v2.20 Index To HelpFile

CT Menu Basics

The most significant features in CTSHELL.INI are the entries that define the menus, the AUTOEXEC section, and the USER options.   They are all similar in construction, they all contain five sets of braces, and look like the following examples:

Generic form: {EntryName} {DirPath} {ExePath} {Switches} {Keyword}

{NOTEPAD Editor} {} {notepad.exe} {!} {}

The name that will show up in the menu is *NOTEPAD Editor*.   Running that program doesn't require changing to a different directory, so the second set of braces is left empty.   The third set of braces contains the name of the program to executein this case, it's *notepad.exe*.   The fourth set of braces contains any switches, or arguments that you need to pass to the program when it runs.   Here, the exclamation point indicates that we want to edit the *current file*, the file that is highlighted in CTShell's list of all files in this directory.   Finally, the fifth set of braces is left empty, because this command doesn't require a special CT keyword.

To edit a specific file every timesuch as to create a menu entry that allows you to edit MYFILE.TXT any time you want toyou would simply replace that exclamation point with the full path name of the file you want to edit:

{Edit CTSHELL.INI} {} {notepad} {c:\windows\myfile.txt} {}

You can send a list of all the *tagged files* to a program that accepts multiple filename arguments by putting an pound sign in that field, such as {#}.

If you need your command line to include the base filename (of the current file) without its extension, you specify that with the commercial at-sign {@}.

KEYWORDS

Many of the features that CT makes available are based on special keywords that take the place of program names.

```
{Set Preferences} {} {} {} {PREFER}
```

CT keywords may be entered in uppercase, lowercase, or mixed case. They are all converted internally to uppercase for evaluation. This example shows how to create a menu entry that would load NOTEPAD, feed it the current file, and install it as an icon, ready to be used at any time:

```
{Edit CTSHELL.INI} {} {notepad.exe} {!} {ICON}
```

DIRECTORIES

Besides running programs, CT makes it easy for you to change to any directory on your disk drive.   An entry that is intended just to change to another directory usually has an entry name in the first field, a directory designation in the second field, and nothing in the remaining three fields:

```
{PIF Directory} {C:\WINDOWS\PIF} {} {} {}
```

CTSHELL.INI is an ordinary ASCII text file, and you can modify it with nearly any editor or word processor, not just with Windows NOTEPAD.   Much of what is found in this file can also be modified from various dialogs that CT-Shell provides.

AUTOEXEC

Towards the beginning of your CTSHELL.INI is an area marked [AUTOEXEC], that allows you to create entries that will load or run programs automatically when CT is started.

These autoexec entries are just like the menu entries. Users often just copy-and-paste from the lower section when they want to start a program automatically when CT starts. You don't really need an entry name for anything used here in the Autoexec section, but it doesn't hurt to leave it there as a reminder for you. CT will simply ignore the first field.

Here is where you're most likely to use the ICON or LOAD keywords:

```
{Calculator} {} {calc.exe} {} {ICON}
```

If you start any additional copies of CT, they will bypass the AUTOEXEC process, so you don't inadvertently start additional unwanted copies of those programs. (See also Load and Run for an alternative.)

LOAD= AND RUN=

CT-Shell will process the LOAD= and RUN= lines that you may already have in your WIN.INI file.   Thus, if you install CT as your primary Windows shell (in place of ProgMan, for example), you'll find that those existing programs are still run at startup, just as before.   (See also Autoexec for an alternative.)

CT enhances those features by allowing up to 512 characters per line, compared to the original 80-character limit.   Thus, if you need to start several programs that require long path designations, you'll have more room for them when they're processed by CT instead of ProgMan.

Note that CT also allows you to install a MAXIMIZE= line and a HIDE= line in your WIN.INI file if you want them.   They work like RUN=, but will start your Windows program in its maximized state or start it as a hidden process.

User Entries

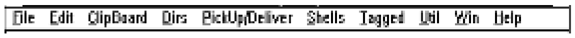
The third place where you'll find entries that look almost exactly like the ones you've been working with so far is in the [OPTIONS] section, in the lines that begin with   User1= ,   User2= , etc.   These are user-supplied function key assignments, and the twelve that you see here become assigned to the function keys F1 through F12, respectively.

```
User1={Toggle Current} {} {} {} {TOGGLE}
User2={Tag All} {} {} {} {TAGALL}
User3={Untag All} {} {} {} {UNTAGALL}
User4={Invert Tags} {} {} {} {INVERT}
     . . .
```

Alarm Entries

At the beginning of the CTSHELL.INI file is a section called [ALARMS], which contains options that control up to five event timers that CT provides. You can configure CT to run a backup program each morning at 3am, for example, or call a remote system and download a packet of email or database files late at night, when the phone rates are lowest. Normally you would change these settings by selecting a menu entry that invokes the special CT ALARM keyword.

MENU


File Edit ClipBoard Dirs PickUp/Deliver Shells Tagged Util Win Help

When CT is run, at the top of its main window is a menu that shows several items, such as *File*, *Edit*, *Clipboard*, *Dirs*, and more. This menu is based on the entries in the CTSHELL.INI file. As you add entries to your CTSHELL.INI file, those new options will appear in the CT menu, and you'll be able to select them from within the program.

FUNCTION KEYS

All of the function keys may be reassigned by the user, to allow you to customize your CT setup as you see fit.   In fact, since you can start additional instances of CT using other xxx.INI files, each instance can map the function keys in its own way.   You change the settings by editing your CTSHELL.INI file, or by invoking the *Preferences* dialog from the *Shells* menu.   In that dialog is a combo box of user commands, which you can edit from within CT.

The following sections describe the default settings for the function keys, based on entries in the supplied sample CTSHELL.INI file.

CURRENT FILE/TAGGED FILES

Much of what CT does with files can be done either with a current file or with a set of tagged files. When a file is tagged, its entry in the files list at the right is highlighted, letting you know that it has been selected for an operation. The first five function keys are devoted to managing those file tagging operations, by default.

F1

Toggle Current File

Toggles the tagged/untagged condition of the current file. If, for any operation, you want to be sure that *no* files are tagged, you can turn off the tag for the current file by pressing <F1>.

F2

Tag All Files

Tags all the files (but not directories or drives) that are in the files listing to the right. You can perform a variety of operations on a set of tagged files, such as to copy them all somewhere, delete them all, etc.

F3

Untag all Files

Untags all the files, regardless of how many were tagged, or how they got that way.

F4

Invert Tags

Inverts all the tags. <F4> will tag all the previously untagged files, and untag the ones that were tagged.

F5

Tag By Name

Tags by name. After you press <F5> you will be presented a dialog box (a question-and-answer panel) that prompts you to enter a filespec for tagging. That filespec may include the ordinary DOS wildcard characters, such as you would use to delete or copy certain files at the DOS prompt. Both the * and the ? wildcard characters work here as you would expect them to.

F6

Original Path

Returns you to the original path where CT was first started. As you work with the program, you will have many reasons to change to other drives and/or directories. <F6> will always return you to your starting point.

F7

Reload Menu

Reloads the menu.   You can easily customize your CTSHELL.INI file with an ordinary text editor.   After you make your modifications, you can simply press <F7> to load the new version, without needing to restart CT.

F8

Print File(s)

Will print a formatted and line-numbered listing of the current or tagged files.   Listings are formatted with a left margin that can be hole-punched. Lines of text can be numbered, as can pages, and at the top of each page can be a header that identifies the file, its creation time and date, and the time and date when it was printed.   Refer to two keywords, *Printer* and *Config*, for more information about setting up your printer driver and changing the format of your file listings.

Pickup/Deliver Files

F9

Pickup Files

Pickup files. Places the currently selected files (possibly a single file) into the Packing List, which allows you to carry those files to another directory. You have the option to *copy* them to that destination and elsewhere, or to *move* them there, which also deletes them from their original location.

F10

Surrender Files

Surrenders the files you are carrying to the current directory. This removes them automatically from your packing list, and assumes that you don't want to copy them to any additional locations.

F11

Lose Files

Loses the files currently in your packing list. Use this option to clear out your packing list after delivering files to all the locations where you want them, or simply to clear it out if you change your mind about moving the ones you've picked up.

F12

Packing List

Displays the current contents of your packing list. While the caption bar at the top of your main CT-Shell window will tell you how many files you're carrying around, and this option will present a list of those files.

Esc

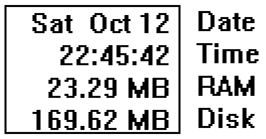
Parent Directory

The escape key is used to change to the parent directory. So that the same operation is easy to accomplish with the mouse, the <Esc> key is represented on the screen along with the function keys.

Command Line

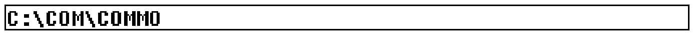
Not a function key, this is a screen-oriented way to open up the CT command line, a place where you can type commands to be run by CT, Windows, or the DOS command processor. Another way to open the command line is with the <Shift+Enter> keypress.

Status Display


Sat Oct 12
22:45:42
23.29 MB
169.62 MB

Below the listing of the function keys is a small window that displays the current date and time, the amount of RAM that is available (including virtual memory if you're running Windows in Enhanced 386 mode) and how much room is left on the current disk drive. The latter two measurements are displayed in megabytes, to the nearest one-hundredth, unless either one drops below one megabyte. If that happens, its display changes to kilobytes instead.

Current Path


C:\COM\COMMO

Just under the menu bar, and above the files display window, is the current path. As you navigate around your disk drive, you can glance here to discover quickly where you are. Watch this as you press <Esc> to move up in your directory tree, and as you press <F6> to return to your starting point.

You can also click the mouse on any part of the path that's displayed, and you'll change immediately to that directory. Thus, you can move upwards in the directory tree by pressing <Esc> to move to the parent directory, or jump directly to a directory that is more than one level higher, by clicking on it in the path display.

Files Window

The display of files contains considerable information that is always conveniently visible.   One of the biggest advantages of a visual shell over an ordinary command line is that so much more information can be made available at all times.

Whenever you perform an operation that results in a change to the information that is displayed in your files window, CT will sense that, and update the listing for you.   If you do something that does not affect any of the information that's displayed here, your files window is left unchangedany selected files will be left selected, for example.

Deleting Files

If you press <Del> , one or more tagged files can be deleted. You are prompted for confirmation before that happens. There are additional deletion options, including the CT keywords *Remove* and *Deldir*. *Remove* is implemented in the sample CTSHELL.INI file in the *File* menu item, in an entry called *RmDir*, like the system command of the same name. It requires that a directory be empty of files and subdirectories before it will remove that directory. *Deldir* is implemented in the *File* menu as the entry called *Kill Directory*.

If you are doing disk maintenance and would like to delete an entire directory full of files (and possibly other subdirectories within it as well), first make sure that the current file is the directory you want to delete, then select *Kill Directory*. You'll be asked to verify deletion of the directory with a dialog box which is worded to get your attention.

Another keyword and command that will delete files is *Shred*. This deletes a file after writing a pattern of bits to every byte of the file that prevents it from being used for anything even if someone manages to undelete it. Shredding a file takes longer than simply deleting one, so you will want to use this keyword only when data security is important, and unauthorized access to the file cannot be tolerated..

The sections that follow each describe one of the components of a line in the files display window, and explain what you can do with that information.

<VOLUME> VOLUMENAME

Each disk drive may have one (and only one) optional volume name that identifies that disk in operations such as a DOS DIR command. If there is a volume label present, it must be in the root directory of that drive.

NAME

Directory names are displayed in uppercase, to distinguish them from file names.   The filename extension, if any,   is included in this field.   In addition, following the directory and file listings, the name field will display the various disk drives that are available on the system.

If you would like to change the name of a file or a directory, you can easily do it with the CT *Rename* keyword.   The files may be ordered in the list based on your choice of four available criteria:

1 file name
2 file extension
3 file date/time
4 file size

The sorting method is an option that you can select from your *Preferences* dialog.   If you choose to sort based on file extension date/time, or size, files will be displayed within each group in file name order.
You can also use the keyboard shotcut keys to change sorting order temporarily.   Press 1 to sort by file name, 2 to sort by extension, 3 to sort by date/time, and 4 to sort by size.   If you want to retain the new order the next time CT is started, use the *Preferences* dialog to save those changes before quitting CT.

Size

The size in bytes of the file is shown here. You are also able to find out how many total bytes are included in a set of files that have been tagged, by using one of the special CT keywords, *Tagged*.

You are also able to discover how many subdirectories, files, and bytes a directory contains, using the CT *Dirsize* keyword.

DATE

The date when the file was last modified (created or updated) is shown using the conventional mm/dd/yy format.   Both the time and the date for a file or a group of tagged files can be changed using the CT keyword, *Setdate*.

To change the date/time for any of the files in the current directory, first tag the one or more that you want to change, then select the menu item *Tagged Files*.   Click on the entry named *Set File Date/Time* to open a dialog box that provides a place for you to enter a new date and time.

TIME

CT displays the file's creation time in its full resolution, which is to within two seconds.   DOS displays only hours and minutes when you use its DIR command, although the number of seconds (to the nearest even number) are stored by DOS in the disk directory.

Sometimes it is important to give a file a date/time that is newer than another file, and there exists on many systems a small utility program whose only purpose is to change a file's date/time to the current date/time.   CT has two keywords that provide this service, called *Touch* (which changes only the current file) and *Ttouch* (to update a list of tagged files).

ATTRIBUTES

The file attributes are displayed as a series of characters which may include any of the letters *RHSDA*, for *R*ead/only, *H*idden, *S*ystem, *D*irectory, and *A*rchive, respectively. These attributes indicate that a file has certain properties which may affect how you and DOS can access and use it.

Read/only

A file with the read/only attribute cannot be modified, overwritten or deleted. DOS simply won't allow the operation to happen, unless the read/only attribute is first removed.

Hidden

Hidden means that a file won't show up in an ordinary DIR command from the DOS command processor, and the DOS COPY command won't copy a hidden file.

System

System means the file is a special type which is part of DOS itself. Examples of this type of file include the two parts of DOS that you'll find in your root directory, named differently depending on which version of DOS you're using.

Directory

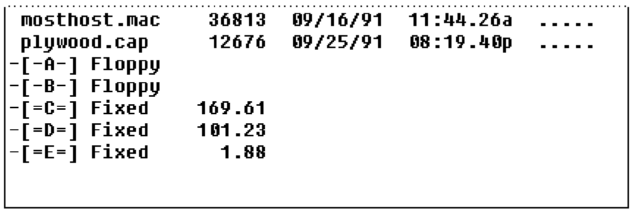
Directory makes the file a subdirectory, rather than a data file or a program. In the DOS system, subdirectories are special files that contain information about the files that are stored under them.

Archive

The archive attribute means that a file has been changed since the last time it was backed-up.

Changing Attributes

Besides displaying the attributes, CT makes it easy for you to change most of them.   You can alter the attributes for a single file or for a group of tagged files with a menu entry that uses the keyword *Attrib*
.
When you invoke that menu entry, CT will present a dialog box that lets you determine which attributes are to be turned on and which ones are to be turned off.   The attributes that you select are not added to the existing ones, but replace the existing ones.   Thus, be sure you select *all* the ones that should apply.   You can turn off all the attributes by leaving them all unchecked, and selecting [OK].



```
 mosthost.mac     36813  09/16/91  11:44.26a  .....
 plywood.cap      12676  09/25/91  08:19.40p  .....
-[-A-] Floppy
-[-B-] Floppy
-[=C=] Fixed    169.61
-[=D=] Fixed    101.23
-[=E=] Fixed      1.88
```

Disk Drive Display

At the end of the files listing, you'll find entries for all the disk drives in your system. Each is identified as to type, and each (except a floppy) has its current remaining capacity.

Extended Selections

CTShell's file window is programmed to allow extended selections. Thus, you can tag multiple files by holding down the <Shift> or <Ctrl> keys as you tag with the mouse. <Shift> will allow you to extend a selection to include contiguous files (a group all together) and <Ctrl> will let you select any files, even if they are separated by others that you don't want tagged.

You can also mark a series of files using the keyboard. Press the key combination <Ctrl+Shift> and move the bar up or down with the keyboard cursor keys. As the highlight bar moves up or down, the files that it passes over will become tagged, just as if you'd dragged the mouse over them.

Doubleclicking Entries

Things happen when you doubleclick the mouse on an entry in the files list, or
move the highlight bar to an entry and press <Enter>:

Directories

If it is a directory, then you will change to that directory. CT lists all the directories first in the file list, to make it easier to travel around your drive by clicking on directory entries. Be reminded that you can use the <Esc> key to change to the parent directory at any time, and that you can click the mouse in the path window (just above the files list) to change to any place in the path above this directory, up to the root directory.

Executable Files

If it is an executable file (to CT, that means .EXE, .COM, .PIF or .BAT) you will execute that file. This provides a convenient way to run programs that are in the current directory, and which require no command-line arguments.

Known Extensions

CT checks the [Extensions] profiles in your WIN.INI file, and can "run" files that are not themselves executable, but for which you have provided an extension association in your WIN.INI file.

CT provides a convenient way for you to modify and add extensions to your WIN.INI file. From the default *Windows* menu, select the *WIN.INI Extensions* entry to invoke a dialog that you can use to discover whether an extension is known, and if so, what program it is associated with. You can also change an existing association or add a new one for an extension not previously found there.

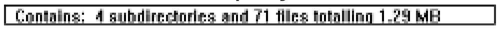
Drive Specifications

If it's one of the entries at the end of the files list that describes a disk drive in your system, doubleclicking on it will change to that drive, which will then become the default, or current, drive. As in many earlier examples, CT will continue to work in the new drive/directory until you change away from it, but you can instantly return to your original startup directory by pressing <F6>.

STATUS LINE

Not to be confused with the *Status Window* to the left, the *status line* just below the files listbox contains a changing stream of information depending on what operation you are currently involved in.

Under certain circumstances, this line contains information about the number of subdirectories in the current directory, and the number of files and their combined size. That's the case when you have first moved to a new directory and haven't yet told CT to do anything else for you. If the status line has been cleared and you would like to see the file size information once again, use the *Update Directory* entry in your *Dirs* menu. That display looks like this:



Contains: 4 subdirectories and 71 files totalling 1.29 MB

THE CT-SHELL COMMAND LINE

CT offers an option, available through the *Preferences* dialog, that allows you to keep the command line open following a command, so that you can continue to process DOS, Windows, or CT-Shell commands without reopening it.

If this option is selected, the command line will go away when you select the [Cancel] button.   If this option is turned off, the command line will close after each time you use it, and you will need to reopen it for any subsequent commands.   Remember, that's easily done with <Shift+Enter>, so the command line is always available quickly whenever you need it.

You can also open the CT command line from the system menuthat little bar-in-a-box in the upper left-hand corner that you can doubleclick to close most Windows programs.   If you first minimize CT into an icon, you'll find that you can leave it in that condition and still open a command line by clicking once on the CT-Shell icon to open the system menu, then selecting the *CT Command Line* entry from that menu.

DOS COMMANDS

Most common DOS commands like CD, RD, MD, COPY, and DEL are handled internally in CT, without using the DOS command processor at all. Most of the DOS commands that CT handles offer an enhancement over their DOS counterparts. For example, the CD command will allow you to specify a drive as well as a directory to change to. RD, MD and DELDIR all allow you to specify multiple arguments. Thus, the command:

```
MD bin init include lib
```

will create four subdirectories in the current directory. You won' t need to issue a separate MD command for each directory you want to create, as you would at DOS.

The COPY command uses a buffer up to 16 times the size of the one DOS uses, allowing many files to be copied with only one disk read and one disk write, for better efficiency.

If you enter a DOS command that CT cannot handle itself, it will pass that command along to your DOS command processor for evaluation. Fortunately, all the most-often-used DOS commands can be dealt with smoothly within CT, without involving the DOS command processor at all. However, if a command involves the DOS redirection operators (<, >, >>) or the pipe operator (|), the whole command is passed to DOS immediately, as is any command to run a .BAT file or a .COM file.

CT Commands

Some additional CT commands may be issued from this command line as well:

Deldir

You can delete a directory, and all the files in it with this command. In fact, it's one of the CT commands that accepts multiple arguments, so if there are a number of subdirectories that you want to remove, you can handle them all with one command.

Detab

This command removes the tabs from one or more text files by replacing them with an appropriate number of spaces. The tab width is dependent on the *TabSize* entry in the [OPTIONS] section of the CTSHELL.INI file.

Encrypt

Changes one or more files into encrypted versions for security purposes. Encrypted data files cannot be read, and encrypted program files cannot be executed. The same command will decrypt an already-encrypted file, returning it to its original condition.

Entab

This command inserts tabs into one or more text files, replacing an appropriate number of spaces. The tab width is dependent on the *TabSize* entry in the [OPTIONS] section of the CTSHELL.INI file.

Find

The *Find* command is designed to find a string (of characters) wherever it may occur within any of the text files in the current directory. Use quotation marks to enclose strings that include embedded spaces.

FormFeed

It is often useful to send a command to the printer that tells it to eject the current page. For example, if you use the *Copy* command to send a text file to your printer, it will probably stop printing somewhere in the middle of the last page. With the *Formfeed* command, you can easily tell your printer to eject that page.

Move

If you want to move a file quickly from one place to another, rather than copying it, you can use the CT *Move* command. The syntax is just like the ordinary *Copy* command, but the move is much faster.

Print

Simulates the DOS PRINT command, by sending one or more unformatted file(s) to the standard printer. Much faster than a formatted print with <F8>.

Shred

This one deletes a file more thoroughly than the *Del* command, making it meaningless even if someone is able to undelete it.   *Shred* takes longer than *Del*, so it should be used only when data security is important.

Where

If you want to know where a file is on the disk, you can use the *Where* command.   The customary DOS wildcard characters are acceptable here, so you could search for files such as:

```
where *.dbf
```

or

```
          where copy??.bak
```

RUN MODE

If you are issuing a command to run a Windows application, you can select from the options in the upper right-hand corner that allow you to run that program as an icon, or full-screen, or normal size.   (Actually, the default is normal, if you don't select either of the others.)

If a PIF file specifies that a DosApp is to be *run in a window*, then the CTShell run mode options will be able to affect it.   Otherwise, only the entries in the PIF file will have any meaning for it.

Command Recall

CT maintains an internal doubly-linked list of previous commands, and lets you scroll through them to select a command to issue again. Each command that you type at the command line is added to the list, and there are three options for deleting old commands that you no longer want to scroll through.

CT-Shell Keywords

All of the internal operations of CT-Shell are invoked using special keywords in the menu entries. Thus, users are free to construct a menu system that can access all of CT's features. Each of the available keywords is explained in this section.

About

Displays information about CT, including the number of the version that you're using.

Alarm

Allows the user to modify the settings used by the five event timers that CT provides. If a time is established and that timer is enabled, a message entered for that timer will be displayed at the specified time, and any menu-type entry placed in the event field will be executed at the stated time. Note that messages can be displayed without starting an event, and events can be started without displaying a message.

Attrib

Changes file attributes. Use this keyword in a menu item to let you change the attributes of the current file. There is another version listed below that changes the attributes for a group of tagged files.

ByName

Presents a dialog box so you can specify a wildcard file name to use for tagging files.

Command

Invokes the command processor that is associated with your COMSPEC environment variable. In most cases this will be COMMAND.COM, the command processor that is supplied with MS-DOS and PC-DOS.

Config

Configures the file listing options. These are the settings within CT that affect how file listings are printed. See also the *Printer* keyword for access to the printer driver itself, which provides control over your installed printer.

Copy

Copies a file to another location. You are prompted for a destination for the current file, and it will be copied to that destination. Like the DOS COPY command, you may supply a file name or a directory as a destination. Like the *Copy* command that you use at the CT command line, this uses a much larger copy buffer than DOS does, for better efficiency.

Note that *you may provide multiple destinations* when this dialog asks you for a destination.

Deldir

Deletes the currently-selected directory and all files in it. Be careful! This one is so powerful that there are *two* confirmations necessary to make it work (you're asked *twice* whether it's okay to delete the directory).

Deliver

This keyword is part of the CT-Shell Pickup/Deliver functionality, and is useful only after you've picked up one or more files using the Pickup keyword. It will copy any files from your current packing list to the current directory. Deliver does not remove those files from your packing list, so you are free to Deliver them to any number of destinations.

Delete

Deletes a file. This removes a file in a way that often cannot be reversed. Be careful, and be sure that you mean it when you use this keyword. You are asked only once for confirmation:

DelLog

When your log file becomes larger than 50 KB in size, CT will call your attention to this fact the next time it enables logging. That occurs each time you start CT, or when you turn logging back on after having shut it off. You are given the option to delete the log file on the spot, or to leave it alone and continue.

Detab

This keyword removes the tabs from one or more text files by replacing them with an appropriate number of spaces. The tab width is dependent on the *TabSize* entry in the [OPTIONS] section of the CTSHELL.INI file.

DirSize

Displays a listing that shows you how big a directory is. It shows how many subdirectories it contains, how many files, and how many bytes they all add up to.

EditIni

If you have provided a default editor name (in your *Preferences* dialog), CT will be able to edit your current CTSHELL.INI file wherever it may be, without your needing to change to that directory and execute your editor explicitly. Note that CT assumes a default editor name of NOTEPAD.EXE if you have not changed it, so this feature should work for everyone, right from the start.

EditLog

Use this keyword to invoke your default editor to view or change your current log file.

Encrypt

Changes the current file into an encrypted version for security purposes. Encrypted data files cannot be read, and encrypted program files cannot be executed. The same command will decrypt an already-encrypted file, returning it to its original condition.

Entab

This keyword inserts tabs into one or more text files, replacing an appropriate number of spaces. The tab width is dependent on the *TabSize* entry in the [OPTIONS] section of the CTSHELL.INI file.

Examine

The *Examine* keyword gives you a chance to see just what contents are in your entry fields at the time CT-Shell is just ready to execute your instructions.

Exit

Shuts down CT-Shell. You can also do this by double-clicking on the system menu box in the upper left corner of CTShell's window, but some people find it easier to pick an exit command out of a menu instead.

Extensions

CT-Shell makes it easy to start a program by doubleclicking on an executable filename in the files listbox, and it extends this capability to non-program file types that have an executable program associated with them. Your Windows WIN.INI file has a section called [Extensions] that lists the file name extensions you want associated with particular executable programs, and CT-Shell provides an easy way for you to view and modify those associations without leaving CT.

FileClip

This keyword creates a file of the current clipboard data, if there is text data there. You are asked for the file name to use, and you may provide a simple filename or a complete path/file name. If you use a simple filename, the file will be created in the current directory.

FileInfo

Copies the descriptive information (from the files listbox) to the clipboard, from where it can easily be pasted into a document with nearly any editor, or used in other programs for which it is appropriate.

Find

The *Find* keyword is an implementation of the same logic that was described in the earlier section about the *Find* command, which can be used at the CT-Shell command line. With it you can find all the occurrences of a string of characters within the files in the current directory, edit one of the entries, print a list of the matching files, or copy a list of matching files to the clipboard.

FullScreen

This keyword causes a program to be run the full size of the display screen, rather than its normal default size. Some programs, such as CT-Shell when used without the *FlexSize* keyword, limit themselves to a fixed size, and will ignore this option.

FormFeed

When you have copied something to the printer, and it has not advanced the last sheet, you can force a formfeed with this keyword. It is invoked by a menu entry in the *Edit* menu of the default configuration.

Help

Runs the Windows help engine. Provides access to this online helpfile that explains what all these options do, and reminds users how to use CT-Shell. You can start at the index, and select the topic you want to review.

Hide

One of the RunMode keywords that specifies how a program is to be run. This is used in the last field of a menu entry, the *Keyword* field, and affects only programs that can be run in this mode. In particular, DOS applications must have a PIF file that runs them in a window, or this will have no effect.

Home

Changes the CT "home" directory to the current directory, so that when you press <F6> this is where you'll return to, rather than the directory in which CT was started.

Icon

One of the RunMode keywords that specifies how a program is to be run. This is used in the last field of a menu entry, the *Keyword* field, and affects only programs that can be run in this mode.

Invert

Inverts the condition of all the tagged files. Those that were selected become unselected, and vice versa.

IsClip

Is there text in the clipboard? If you want to be sure before printing the clipboard, or before starting another application to use that data, you can test with this keyword.

Load

One of the RunMode keywords that specifies how a program is to be run. This is used in the last field of a menu entry, the *Keyword* field. This one causes a program to be run as an icon, and is a synonym for the *Icon* keyword (above).

Lose

When you have used *Pickup* to collect some files to be delivered elsewhere, they will remain onboard until you either *Surrender* them to a directory or *Lose* them. This keyword removes all the files from your packing list, so that you will no longer carry them around with you.

Mail

Displays a message box that tells whether there is mail waiting to be read, or waiting to be sent to a destination.

Maximize

One of the RunMode keywords that specifies how a program is to be run. This is used in the last field of a menu entry, the *Keyword* field, and affects only programs that can be run in this mode. This one causes a program to be run full-screen, and is a synonym for the *FullSize* keyword (above).

Move

Moves a file to another location. If the destination is on the same drive as the source file, this feature changes the directory information relative to a file without copying the file itself. Thus, a move from one directory to another on the same drive takes only part of a second, no matter how big the file that is being moved. No file data needs to be read or written, just the directory entry for that file.

If this keyword is used to move a file across drives, CT will copy the file, verify that it arrived safely, and delete the original. Although that takes longer than moving a file to another directory on a single drive, the functionality remains the same.

MoveHere

You may want to *Pickup* files from one or more locations, then *Move* them to the current directory, rather than copying them to here with *Deliver* or *Surrender*. This keyword provides that service. It also removes those files from your packing list (since the original path/file names are no longer valid), so there is no need to *Lose* them after *MoveHere*.

Normal

One of the RunMode keywords that specifies how a program is to be run. This is used in the last field of a menu entry, the *Keyword* field. This option is the default, if no other RunMode is specified in the last set of braces. As such, it is never actually needed, though some users may like to use it to document their entries explicitly. It is a synonym of the keyword *Run*, and it is the RunMode used whenever CT-Shell processes programs found in the RUN= line in your WIN.INI file.

OrigPath

Returns to your original path, or one that you have explicitly changed your original path to be. This is invoked by <F6> in the default setup, and provides a way easily and quickly to return to the drive/directory from where you started CT-Shell.

Packing

Displays a packing list, showing you all the files that you've picked up with the *Pickup* keyword, and haven't yet gotten rid of with the *Surrender*, *Lose* or *MoveHere* operations. You'll know that you're carrying files around with you by the appearance of the caption at the top of your CT-Shell window. To find out exactly what files they are, invoke this keyword.

Phone

The built-in phone directory can be used to recall numbers for manual dialing, and on a computer that is equipped with a modem, it can be used to automatically dial a listed number. Although a large number of phone numbers are more easily added to the CTSHELL.INI file with an editor, the dialog that is invoked by the *Phone* keyword can also be used to add and delete entries. As well, it is used to look up numbers and optionally to dial them.

Position

CT-Shell defaults to displaying itself centered on the screen. Its main window is small enough to fit well using the resolution of any monitor that can be used with Windows.

Those with higher-resolution monitors, such as 800x600 and above, will have considerable choice regarding where to locate CT. By grabbing the caption bar at the top of the window with the mouse cursor, you can hold down the left button and drag CT to the location you prefer. Having done so, you can invoke the *Position* keyword by selecting the appropriate entry from the *Shells* menu, and thereby establish a new starting position for CT.

Prefer

This one invokes the *Preferences* dialog that makes it easy for you to change many of the settings that are stored in your CTSHELL.INI file, without needing to load and edit that file with an ordinary editor.

Print

This begins the process of printing one or more selected files. Printed copies of text files are usually called *listings* when they do not refer to documents, such as letters.

CT offers a wide variety of options to control the way those listings are created, allowing you to choose headings, page numbers, line numbers, and more. See the description of the *Config* keyword, above, and the discussion of the printer options in the manual for more details about these options.

PrintClip

If there is text currently in the clipboard (see the *IsClip* keyword), you can print a copy of it with this option. Line size and whether to use fixed-width text is determined by your current printing options, which you can change by invoking the *Config* keyword.

Printer

Invokes your printer driver setup function. The exact set of features and options that are offered by this function depends on the printer driver that you use. This is probably where you can change from portrait to landscape mode, determine how high your graphic resolution should be, download font software, etc.

PrintList

Prints a copy of the *selected* files in the CT-Shell files listbox. Note that this is handled differently than the *Where* and *Find* listboxes, where all the entries are included when you ask to print them.

PrintUnf

Prints the current file in an unformatted fashion by copying it to the standard printer. This is much like the DOS PRINT command, and doesn't provide a formatted listing, but is quite fast by comparison.

Reboot

There may be times when you have made changes to a system configuration file (such as CONFIG.SYS or AUTOEXEC.BAT), and you need to reboot your system for those changes to take effect. The *Reboot* keyword does that from within CT-Shell, and is different than pressing <Ctrl+Alt+Del> in an important way. By invoking this keyword, you give Windows applications a chance to shut down in an orderly way before the rug is pulled out from under them, so to speak.

Reload

If you have edited your CTSHELL.INI file manually, rather than changing its contents with the various dialogs that CT-Shell provides, you will need to let CT know that you want it to re-read the setup information and reconfigure itself. *Reload* does just that. In the default configuration, this is assigned to the <F7> key.

Remove

Removes the current directory, essentially duplicating the operation of the DOS command RD (or its synonym, RmDir). Like the DOS command, *Remove* requires that a directory be empty before it can be removed.

Rename

Changes the name of a file, and unlike the DOS REN command, CT will change the name of a directory as well.   This is actually implemented as the same low-level DOS function that MOVEs a file, and it can be used for the same purpose.   If you provide a new pathname that includes a different directory than the current directory, your file will be not only renamed, but moved to that directory as well.

Restart

After making major changes to any of the Windows configuration files (such as WIN.INI, SYSTEM.INI, etc.), you may want to close CT-Shell and restart Windows, so those changes can take effect.   This keyword will restart Windows, after giving your applications a chance to shut down normally. (See the discussion of *Reboot*, above.)

Run

One of the RunMode keywords that specifies how a program is to be run. This is used in the last field of a menu entry, the *Keyword* field, and affects only programs that can be run in this mode.

This is a synonym for the *Normal* keyword, and as such, it is the default if no other RunMode is specified.   This is the mode that CT-Shell uses if it processes any programs in the RUN= line of your WIN.INI file.

SetDate

Displays a dialog box that allows you to change the date/time for one or more selected files.

Shred

Destroys a file so that it can't be read even if it is somehow undeleted.   This is analogous to feeding a document into a paper shredder, for security reasons.   The confirmation dialog looks almost exactly like the one for the *Delete* keyword.

SysDate

This one lets you reset the system date and time, that is, the computer's main clock.   It works like the DOS commands DATE and TIME, but it does it from within CT-Shell.

Surrender

Having picked up one or more files from one or more directories using the *Pickup* keyword, the commonest subsequent operation is to copy them somewhere else, and empty them from the packing list of files that you were carrying with you.   The *Surrender* keyword handles both these tasksthe copying and the emptying.   It is a combination of the routines *Deliver* and *Lose*, which are both available to be used separately.

System

Displays system information. This is similar to the information you can get from the Windows Program Manager by clicking on its HELP/ABOUT option. You can find out what mode you are running, using what kind of processor and coprocessor (if any), and whether small-frame or large-frame EMS operation is in effect (if any). This keyword does not display the amount of memory available, as CT displays that at all times anyway:

TagAll

Tags (selects) all the files in the current directory. By default, this is assigned to the <F2> key.

Touch

Makes the date/time of the current file reflect the current date/time. This is mainly of interest to programmers, who sometimes need to adjust a date in this way while using a MAKE type program maintenance utility.

Untag

Untags all the files in the current directory. The default configuration assigns this operation to the <F3> key.

Uninstall

Removes CT-Shell as the default Windows shell, and replaces the shell that was in use when CT was first installed.

Where

The *Where* keyword is an implementation of the same logic that was described in the earlier section about the *Where* command, which can be used at the CT-Shell command line. With it, you can locate a file anywhere on the current disk drive, go to that location, print a list of all the matching files, or copy a list of matching files to the clipboard.

Tagged

Displays the number and size of tagged files. If you have tagged a set of files to be copied to a floppy disk, you might want to check to be sure that the number of bytes tagged does not exceed the number of bytes that are free on your disk.

Tattrib

Changes the attributes of tagged files. If you should want to change all the .EXE and .COM files in a directory to read/only status, to prevent unnecessary share violations with a network, you could tag those files, then use this keyword to give them all a read/only attribute.

Tcopy

Copies a set of files to another location. You must provide a directory as the destination. CT does not support file concatenation (combining several files into one) by copying multiple files to a single file. However, CT does support multiple destinations for a multiple file copy. Put another way, you can tag an assortment of files that you want to copy to two places, then when the dialog asks for the destination(s), you can provide both. When the first series of copies has been made, the second will be done automatically.

Tdelete

Deletes a set of tagged files. You are prompted for confirmation before the deletion is accomplished:

Tdetab

Removes the tabs from a series of tagged files by replacing them with an appropriate number of spaces.

Tencrypt

Encrypts a series of tagged files, rendering them unusable for their original purposes. Using this keyword on already-encrypted files returns them to their original condition.

Tentab

Inserts tabs into a series of tagged files, each of them replacing an appropriate number of spaces.

Tmove

Moves a set of tagged files to another location. Just as with the single file move keyword, if the move is from one place to another on the same disk drive, these files are not physically copied to their new location, just their directory entries are changed. If the move is done across drives, a copy is first done, and the original file removed after verification that the copy was successful.

TprintUnf

Copies a set of tagged files to the standard printer, with a formfeed following each. Much like the DOS PRINT command, this is much faster than creating a formatted listing.

Tshred

Shreds a set of tagged files, so that their data is unrecoverable even if the files themselves are undeleted.

Ttouch

Changes the date/timestamp of all the tagged files to the current date/time. Used mainly by programmers who use a MAKE type program maintenance utility.

OBJECT-ORIENTED SUBSTITUTION

Here is a listing of all the special field characters that may be used in CT pop-up menu entries. Although any of them may be used in any of the fields except the entry name field and the keyword field, you will find that certain ones are likely to be used in the directory path field, and other ones are more likely to be of use in the executable path and switches fields.

In each of the following examples, a DOS command line is shown, to illustrate how the command would look if it were entered normally at a DOS prompt. After that, the CTSHELL.INI entry is shown that would produce that command, substituting current information for the CT special characters. To see the examples as they were intended, you may need to enlarge your help window.

!

The exclamation point translates into the current file name. Here's an example that would use the Windows NOTEPAD.EXE editor to edit the current file:

```
Command line:    notepad.exe filename.ext

CT entry:     {&Edit} {} {notepad.exe} {!} {}
```

~

The tilde translates into the current file name, but including its full path. Here's an example that would use the Windows NOTEPAD.EXE editor to edit the current file, assuming that C:\FOO\BAR is the current working directory::

```
Command line:    notepad.exe c:\foo\bar\filename.ext

CT entry:     {&Edit} {} {notepad.exe} {~} {}
```

#

The pound sign translates into a list of files that are tagged, or as many of them as can be squeezed into the DOS limit of 127 characters on a command line. You might like to add all of the tagged files to an archive file named ARCNAME.LZH by using the LHArc program:

```
Command line:     lha a arcname file1 file2 file3 ...

CT entry:     {&LHArc Add} {} {lha.exe} {a arcname #} {}
```

If you give CT a command this way that turns out to be longer than the allowable 127 characters, it will let you know about it and offer an opportunity for you to abort the operation or truncate the command to less than 127 characters and go ahead with it.

@

The commercial at sign translates to the root filename of the current file, without any extension. For example, if the current file were named FOO.EXE, then the {@} in a command would become FOO.

There aren't many cases when the root filename is needed, and programmers will probably find this one more useful than most other users. For example, if the current filename is FOO.EXE, the expression {@.C} would translate to FOO.C, and the expression {@.EXE} would translate to FOO.EXE.

If you do store all your .PIF files in a special subdirectory, you can create a menu entry that will allow you to edit the PIF file for any .EXE that happens to be the current file:

```
Command line:    pifedit.exe \pif\filename.pif

CT entry:    {&PIFedit an EXE} {} {pifedit.exe} {\pif\@.pif} {}
```

?argument?

A pair of question marks surrounds the prompt you want CT to display when it asks you for a string of characters to put in its place. This is how you can supply variable arguments at the time an entry is executed.

For example, the LHArc command shown earlier will always create an archive called ARCNAME.LZH, because the name ARCNAME has been *hard coded*, or stated explicitly, in the command. It will require that the same archive be created or updated each time this pop-up entry is executed, although the currently tagged file names may be different each time. Compare that to this example, where CT will ask the user for an archive name each time the command is executed:

```
Command line:    lha a arcname file1 file2 file3 ...

CT entry:    {&LHArc Add} {} {lha.exe} {a ?Archive Name? #} {}
```

When this pop-up entry is executed, CT will display a dialog box identifying the needed argument as "Archive Name" and asking the user to supply a name. That answer will be inserted into the command line, replacing the ?Archive Name? characters.

%variable%

A pair of percent signs will cause CT to insert the value for a named environment variable into the command. This is consistent with the way environment variables can be accessed within a batch file, and the topic of environment variables is explained fully in your DOS manual.

Briefly, you set environment variables to a given value with a SET command like this:

```
SET ENVAR=contents of variable
```

Although that can be done at a DOS prompt, it is usually done in an AUTOEXEC.BAT file instead, so that your environment variables are established correctly each time you start your computer.   Programs can obtain various kinds of information from environment variables, and the documentation for those programs will tell you how to set them, if any are needed.

; Comment

If you want to place comments in your [ITEMS] section, all you need to do is keep them outside of braces.   However, if you'd like to temporarily change an entry to a comment (such as to test CT without that entry), you can place a semicolon in the first column on that line.   Removing the semicolon later restores the meaning of the line.