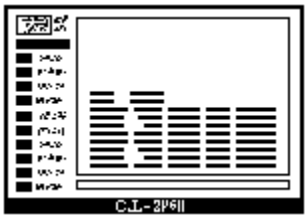
CT-Shell for Windows

v2.00



Index to CT-Shell Help

Multiple Configurations

Note that no matter how or where you start CT, you may provide an optional initialization filespec on the command line after the program name. Rather than looking for its default initialization file–CTSHELL.INI–in any of the places where CT would normally expect to find it, it will instead use the name you provide as an initialization filespec.

This makes it possible to set up additional configurations for specific purposes, such as programmand or word processing. Also, multiple users of the same computer can each have their own custom configurations.

CT Menu Basics

The following few sections provide an overview to let you know how simple that system is to set up the way you want it. Every menu entry is based on your CTSHELL.INI file.

The CTSHELL.INI file also contains values for many options that CT relies on for its various features. 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: {Entry Name} {DirPath} {ExePath} {Switches} {Keyword}

Example: {NOTEPAD Editor} {} {notepad} {!} {}

The name that will show up in the menu is *NOTEPAD Editor*. Running that program doesn't require changing first 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 execute–in this case, it's *notepad*. The fourth set of braces contains any switches, or arguments that you need to pass to the program when it runs.

Here, the exclamation mark 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.

In addition to what you've seen so far, 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 {#}. (Some people call that a *number sign*–musicians call it a *sharp*. It's the <Shift+3> key on your computer keyboard.

Finally, if you need your command line to include the base filename (of the current file) without its extension, you can specify that with the commercial at-sign {@}. Some users will appreciate that they can create an expression that has the same name as the current file, but a different extension.

Keywords

Many of the features that CT makes available are based on special keywords that take the place of program names. Most entries that use a keyword do not use any of the other fields, except the first. This one would activate the dialog that allows you to change your preferences–those settings that affect how CT does certain things:

Example: {Set Preferences} {} {} {} {PREFER}

The special CT keywords may be entered in uppercase, lowercase, or mixed case. They are all converted internally to uppercase for evaluation. There are 30 CT keywords in this version of the program; they are all described in detail in Chapter 6 of the manual.

There are just a few CT keywords that are, in fact, used with executable programs. They include ICON and LOAD, which are synonyms and would cause the program to be run as an icon. FULLSIZE causes a program to be run in the maximized. RUN causes a program to be run normal size. For completeness, the keyword NORMAL is also included, although that's the default if none of the others is used.

Items

Most of the entries in the sample CTSHELL.INI file use CT keywords, since they will work exactly the same way on everyone's computer. However, there are a few entries in the sample CTSHELL.INI file that run some of the Windows utilities.

Directories

Besides running programs, CT makes it easy for you to change to any directory on your disk drive. There's an example provided that changes to your Windows "home" directory. 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.

Example: {PIF Directory} {C:\PIF} {} {} {}

User Function Keys

There are pre-assigned meanings for many of the function keys, but <F9> through<F12> have been kept available for the user to define. In the sample configuration file, they are assigned to several configuration options. You can reassign these four keys to any entries you want.

Here are the default settings for the user function keys, based on entries in the supplied sample CTSHELL.INI file:

F9 Alarm Settings

Allows you to set up to four internal timers that can pop up reminder messages at preset times, or even execute programs according to an established schedule. Check the ALARM keyword when you need to know more.

F10 Listings Config

This will allow you to set all of the options that have to do with printing file listings. You are able to set preferences like the number of characters that a line must accommodate, whether to use a fixed or proportional font, whether headings are to be printed, line number and page numbers, and more.

F11 Preferences

Here is where you can choose many of the options that control CTSHELL. Many of the entries in the CTSHELL.INI file can be set by activating this keyword. For more details, see the keyword PREFER .

F12 Printer Setup

Pressing this one executes the "setup" function from the printer driver that you have installed. This is the same setup function that is invoked by your word processor when you change printer settings there. More details are available where PRINTER is explained.

Default Function Keys

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.

F1 Toggle Current

Toggles the tagged/untagged condition of the current file. If, for some reason, 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

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

F3 Untag All

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

F4 Invert Tags

Inverts all the tags. You might want to tag some of the files in the current directory, copy those tagged files to a floppy disk in drive A:, then copy the rest of the files in that directory to somewhere else. <F4> will tag all the previously untagged files, and untag the ones that were tagged.

F5 Tag By Name

Tags by name. If you want to copy all the .EXE and .COM files from the current directory to a floppy disk, you could press <F5> once, specify *.EXE when CT asks you for a wildcard filespec, finish the process, then do it again and specify *.COM.

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.

There is a CT keyword called HOME that makes the current directory your CT "home" directory, rather than the one where you started the program. If you work centers around one directory for a while, then changes to another location, you might want to change the place CT calls home. The HOME keyword is implemented in the *Directories* menu, in the default configuration.

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 press <F7> to load the new version, without needing to exit CT and restart it.

F8 Print File(s)

Will print a formatted and line-numbered listing of the current or tagged files.

F9 **through**
F12

Are reserved for the user. In the sample CTSHELL.INI file they are assigned as mentioned above, but you can easily reassign them to other tasks that you want to be able to invoke with a keypress.

Esc Parent

The escape key is used to change to the parent directory.

Command Line

Not a function key, this is a screen-oriented way to open up the CT command line. Another way to open the command line is with the <Shift+Enter> keypress.

Status Display

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 hundredth, unless either one drops below one megabyte. If that happens, the display changes to kilobytes instead.

Current Path

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.

The largest window contains a display of the files in the current directory. Information displayed for files includes name, size in bytes, last modified date, last modified time, and attributes.

At the bottom of the list of files is the disk drive display. The CT-Shell files window is designed to allow extended selections, and various things can happen when you doubleclick entries in this listing, depending on the type of entry. CT-Shell makes it very easy to delete files, for convenient disk maintenance.

Deleting Files

If you press <Del> , the current file or an entire set of tagged files can be deleted. You are prompted for confirmation before that happens, of course! There are additional deletion options, including a CT keyword called DELDIR. Another CT keyword that you may find useful for this purpose is SHRED. This one deletes a file, but first it overwrites it with a repeated pattern of bytes that makes it totally meaningless, even if someone is able to undelete it. SHRED takes longer than DELETE, but provides ultimate security..

The following sections summarize the parts of each line in the files listing:

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.

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. This keyword is implemented in the sample CTSHELL.INI file in the *Tagged Files* menu item, and called *Set File Date/Time*.

Time

CT displays the file's creation time in its full resolution, which is to within two seconds. CT has two keywords that can change the file time to the current time, called TOUCH (which changes only the current file) and TTOUCH (to update a list of tagged files). They are both assigned to menu entries named *Touch*, in the appropriate pop-up menus in the sample CTSHELL.INI file.

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. Following this listing of the attributes are directions showing how you can use CT to change file attributes.

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.

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 if your CTSHELL.INI file contains a menu entry that uses the keyword ATTRIB (see the later section on CT keywords for more information about ATTRIB and other CT keywords).

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 floppies) has its current remaining capacity. You can change to another disk drive by doubleclicking on its entry here.

Extended Selections

CTShell's file window is programmed to allow extended selections. Thus, you'll find that 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! (You may also move the highlight bar to an entry and press <Enter>.) When you do either, what happens will depend on what kind of file is selected:

Directories

If it is a directory, then you will change to that directory.

Executable Files

If it is an executable file (to CT, that means .EXE, .COM, .PIF or .BAT) you will execute that file.

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 in your WIN.INI file. Thus, it is likely that if you doubleclick the mouse on a .WRI file, you'll start up Windows Write and can edit that file. If you doubleclick on a .CRD file, you'll start up the Cardfile database program, etc.

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.

The CT-Shell Command Line

From the command line, you are able to invoke DOS commands and special CT-Shell commands. You can set the run mode of WinApps that you run, and you're able to recall past commands that you've already entered.

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.

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.

Find

It often happens that someone needs to edit one of the many text files that are part of a programming or word processing project, and can't remember for sure which file contains the text. The FIND keyword is designed to find a string (of characters) wherever it may occur within any of the files in the current directory.
Use quotation marks to enclose strings that include embedded spaces.

FormFeed

It is quite 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.

SHRED

Deletes one or more files, but first overwrites them with a pattern of bits that renders them useless, even if someone manages to undelete them somehow. This is used where security is important, and a file that's been shredded is totally secure.

Because there is no recourse to this command–no way to undo its effects–it does not accept wild card arguments on the command line. You can shred a number of files with one command, but you must name them all explicitly

Where

If you want to know where a file is on the disk, you can use the WHERE command.

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.)

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.

After one command has been given at the command line, you'll see a [Delete] button the next time you invoke the command line. That will allow you to delete the earlier command.

After more than one command has been issued, you'll see options that let you delete from the current command upwards, from the current command downwards, or just the current command itself. The default is always to delete just the current command, so you don't accidentally remove several that you'd like to use again later.

CTSHELL.INI Reference

Here is information about the various sections in the CTSHELL.INI file, beginning with accessing CTSHELL.INI. The sections you'll find in that file include:

[Alarm]
[Autoexec]
[Color]
[Editor]
[Modem]
[Options]
[Phones]
[Printer]
[Items]

Accessing CTSHELL.INI

Your CTSHELL.INI file is an ordinary ASCII text file that can be edited with nearly any editor or word processor. Although it isn't a powerful editor, the Windows NOTEPAD editor is fine for the light-duty work of customizing your initialization file.

After editing your CTSHELL.INI file, you can put your menu entry changes into effect afterwards by pressing the <F7> function key, or by exiting and restarting CTSHELL.

Be aware that there are several CT dialogs that also affect portions of your CTSHELL.INI file, especially those sections that affect the way options are implemented (as opposed to menu items and entries). Note that any of these other option settings can always be changed by editing CTSHELL.INI, but usually the dialogs provide the easier way to make such choices.

For example, the way your printed listings are formatted is affected by a dialog that is invoked by the CONFIG keyword. The alarm/event settings are affected by a dialog that is invoked by the ALARM keyword. The PREFER keyword invokes a dialog that allows you to set a great number of preferences, all of which will be explained in the sections that follow.

[Alarm]

This section contains the controlling data for up to four event timers that work like alarm clocks. The default CTSHELL.INI file does not contain data for any events or timed messages. The ALARM keyword invokes a dialog from which these controls can be set.

[Autoexec]

This section contains a series of entries (in the usual menu entry format), that determine the programs that will be started automatically when CT is first started.

[Color]

There are three entries here, named as shown below, which each contain one component of the background color that's used for CTShell's main window. These values are among those that can be changed via the *Preferences* dialog, which is invoked by the PREFER keyword.

[Editor]

The section marked [EDITOR] contains two settings that tell CT whether you have a text editor and whether it has a particular capability. These settings are used in conjunction with the FIND command, and will allow you to edit the file that contains a string of characters that you have asked CT to find for you. These settings look like this in the sample CTSHELL.INI file:

```
[EDITOR]
EditorName=NOTEPAD.EXE
EditLine=
```

The EditLine entry tells CT two things about your editor: whether it can start at a line number that is included as part of its command, and if so, what command-line switch is used to invoke that feature.

[Modem]

One of CTShell's functions is to store phone numbers in a dialing directory, and optionally to dial phone numbers for you, using a modem. There are several entries in this section that can be changed, if necessary, to accommodate unusual or nonstandard modems, or phone systems that require additional characters to be dialed to access an outside line or to charge the calls to a credit card.

The sample CTSHELL.INI file includes most of these settings, which will be satisfactory for a large number of users:

```
[MODEM]
ModemInit=ATQ0M1L0V1X4&C1&D2
ModemSpeed=1200
ComPort=1
DialPrefix=ATDT
DialSuffix=
```

The one that is *not* included is the modem initialization string–the first field shown above. If you want to activate this feature for your version of CT, you must have a modem connected to your computer, and you must provide an appropriate modem initialization string. If you're not sure what your modem needs in an initialization string, the one shown here is a good place to start experimenting.

[Options]

Here is a place to set a variety of options that affect the way CT works. Nearly all of these entries may be changed via the *Preferences* dialog, which is invoked by the PREFER keyword. The default CTSHELL.INI file contains the following entries in its [OPTIONS] section:

```
[OPTIONS]
KeepOpen=0
RequireConf=0
IgnoreDrives=
Mail_In=
Mail_Out=
User1={Alarm Settings} {} {} {} {alarm}
User2={Listings Config} {} {} {} {config}
User3={Preferences} {} {} {} {prefer}
User4={Printer Settings} {} {} {} {printer}
```

The entries for KeepOpen= and RequireConf= are TRUE/FALSE type entries, in which a 1 represents TRUE and a 0 represents FALSE.

KeepOpen

KeepOpen refers to the command line, and whether it should be kept open following a command, so that additional commands may be entered without reopening the command line dialog.

RequireConf

RequireConf refers to closing CT by doubleclicking on the system menu box. This option may be changed from the *Preferences* dialog.

IgnoreDrives

If you have a system in which there are a great number of drives available–such as in a network–and you do *not* need a continuing display of the available space on those drives, you may instruct CT to ignore certain drives. The example shown here would cause CT to ignore drives from F: on up:

```
IgnoreDrives=FGHIJKLMNOPQRSTUVWXYZ
```

MailIn and MailOut

One of CTShell's keywords is named MAIL, and invoking that keyword in a menu entry causes CT to look for the presence of files that are identified by these MailIn= and MailOut= entries. It then presents a message box that states in simple *Yes/No* terms whether mail is waiting to be sent (MailOut), or mail is waiting to be read (MailIn).
These path strings may both be modified using the *Preferences* dialog.

User1 through User4

The User1= through User4= tasks are originally assigned to the special CT keywords, ALARM, CONFIG, PREFER and PRINTER. Doing that provides quick

and easy access to these configuration dialogs. These settings may be modified through the *Preferences* dialog.

StartX and StartY

If you invoke the CT keyword POSITION, these entries will be made in your CTSHELL.INI file for you. If they exist, CT will use them as the position of the upper left corner when the program is started.

[Phones]

The entries here are made available to you in the dialog that is invoked with the PHONE keyword. There is a menu entry for that purpose in the default configuration, in the *Shells* menu.

The approximate size of the entries will become apparent when you first invoke that menu item. Subsequent phone numbers are put into place as Phone1= , Phone2= , Phone3= , etc. Legitimate characters for the phone number include the digits, hyphen, parentheses, and the period.

[Printer]

The [PRINTER] section controls certain options that affect the way CT prints a file (or a list of tagged files) when you press the <F8> key. All of these settings are controlled from within CT, from a dialog box that is presented when you execute a pop-up menu entry that uses the CT keyword CONFIG. In the default configuration, that keyword is bound to the <F10> function key.

LineSize

The LineSize= entry will be 80, 110 or 132, if it was entered from within CT, and it specifies the width of text file lines, as you usually work with them. When printing text files, CT will choose the largest font that is available for your printer that will display at least that many characters on a single line, in addition to allowing room for borders and optional line numbers.

Headings

Headings= determines whether name/date/time headings will be printed at the top of your listings.

Pagenums

PageNums= determines whether your listings will have page numbers at the bottom of each page.

24Hour

This option lets you determine whether the times displayed in file listings will be in 24-hour "military" time, or in the conventional AM and PM format.

Linenums

If the file being printed is a program listing, chances are you'll want the lines to be numbered. This option will cause them to be numbered from 1 to 99999, and separated from the text with a > and a space.

Independent

The Independent= option refers to page-numbering for multiple-file printing jobs. If you have a series of files tagged before you press <F8>, all of them will be printed, not just the current file. If Independent=1, each file listing will begin with page 1. If Independent=0, the whole series of files will be page-numbered consecutively, straight through.

Draft

Draft= tells Windows whether to try to find a font for high-quality output or one that will print more quickly, with lower quality. If Draft=1, lower quality will be allowed.

TextFixed

Whether the text portion of your printout is printed in a fixed font or a variable (proportional) font is controlled by this one.

[Items]

This part of your CTSHELL.INI file determines the contents for your main menu, and the pop-up menus that its items invoke.

Menu Items

Each menu item is distinguished by the special word *Item* that appears first on its line, then a set of braces containing the item name as it should appear in the main menu. Since braces are used as delimiters, the menu entries can contain quotation marks, if you want, as well as spaces, parentheses, and most other punctuation.

item {ItemName}

ItemName

The menu item name may contain embedded spaces, and it may contain the special ampersand character (&), which determines a letter that will appear underlined in the menu itself.

Pop-up Entries

Each pop-up entry contains five fields, delimited by braces, of which only the first is required.

Entry Name

{EntryName} {DirPath} {ExePath} {Switches} {Keyword}

The entry name is displayed in the pop-up menu to allow the selection of this option. Like the menu item name, the entry name may contain an ampersand character to determine the character that will be underlined in the menu, thus providing easy access to this item with a keyboard command.

The entry name may also be preceded by a dollar sign ($), which has a special meaning. Such an entry name is always followed by four more sets of empty braces (or at least, CT ignores anything in them), and is used only as a *label* for a section in the menu.

Two other special characters may be used here, to provide separation from other pop-up menu entries. If an entry name begins with a hyphen (-), there will be a horizontal bar in the menu, separating that item from the ones that preceded. If there is a plus sign (+) before the name, that entry will begin a new column in the pop-up menu, with a vertical bar separating it from the preceding entries.

DirPath

{EntryName} **{DirPath}** {ExePath} {Switches} {Keyword}

The directory path is an optional field which, when provided, causes CT to

change either temporarily or permanently to that directory, before executing any program that may be part of this entry.

ExePath

{EntryName} {DirPath} **{ExePath}** {Switches} {Keyword}

The executable path is the path name for an executable file which is to be run when this entry is selected.

Switches

{EntryName} {DirPath} {ExePath} **{Switches}** {Keyword}

Programs often require additional information on the command line when they are run. An editor, for example, can often be told what file to edit by including the file name as part of the editor command.

Keyword

{EntryName} {DirPath} {ExePath} {Switches} **{Keyword}**

Many of the operations that CT performs are handled internally by CT itself. That's how it is able to improve on many of the DOS commands, rather than passing the commands along to DOS. A keyword command is almost always used in place of–rather than in addition to–an executable path.

Keyword Reference

CT-Shell's keywords operate on both single and tagged files. This section includes brief descriptions of all 30 of them. See the manual for more comprehensive information.

Single and Tagged Files

When a keyword is designed to affect a single file, CT needs only to inquire which file is the *current file* in the file list, and apply that operation to that single file. When a keyword is designed to affect a list of tagged files, CT needs to go through *all* the entries in the file list and ask of each, "Is this one tagged?" Then it applies the operation to the ones that are, skipping the ones that are not. Obviously, it is easier and quicker to affect only the single current file.

Keywords that are intended for use with tagged files will almost always work with the current file, which is almost always considered to be tagged.

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 four 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.

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).

Delete

Deletes a file. This removes a file in a way that cannot usually be reversed. Be careful, and be sure that you mean it when you use this keyword.

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.

Exit

Shuts down CTSHELL. 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.

FileInfo

Copies the descriptive information (from the file list window) 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.

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. This provides access to the online helpfile that you are looking through right now.

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.

Mail

Displays a message box that tells whether there is mail waiting to be read, or waiting to be sent to a destination. This feature is actually only commenting on the existence of the two files identified by path names in the MailIn= and MailOut= entries in [OPTIONS] section of the CTSHELL.INI file.

Move

Moves a file to another location. This feature changes the directory information relative to a file without copying the file itself. Thus, a move takes only part of a second, no matter how big the file is that is being moved.

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.

Position

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 point for CT whenever it starts.

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.

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.

Rename

Changes the name of a file, and unlike the DOS REN command, CT will change the name of a directory as well.

Shred

This one deletes a file in much the same way that DELETE does, but first it changes its contents to a pattern of bits that renders the file meaningless even if someone later manages to undelete it. Use this when security is important. It is slower than the DELETE keyword, because of the extra work that's involved.

System

Displays system information. This is the same 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). Also displays the percentage of currently free system resources.

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.

Touch

Makes the date/timestamp 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.

SetDate

Changes the date/time stamp that DOS has applied to a file or a set of files.

Tattrib

Changes the attributes of tagged files. The dialog for this keyword looks exactly like the one that is used to change the attributes of a single file.

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.

Tdelete

Deletes a set of tagged files.

Tmove

Moves a set of tagged files to another location. Just as with the single file move keyword, these files are not physically copied to their new location, just their directory entries are changed.

Tshred

Shreds a set of tagged files, destroying them as well as deleting them. See the SHRED keyword for more details.

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

Certain objects (characters like ! and #) that you place in the command automatically take on values that represent file names or other information.

!

The exclamation mark translates into the current file name.

#

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.

@

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, the the {@} in a command would become FOO.

?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.

%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.