

Cueshell

Desktop Program

User Manual

Albin Hessler Software

AHS

**Albin Hessler Software
Im Zeilfeld 25
D-72631 Aichtal**

© 1994-1996 Albin Hessler Software

AHS

Table of Contents

Introduction	4
The pointer sprite	4
HITs and DOs	4
Keyboard control.....	5
License Agreement.....	6
Limited warranty	6
Software updates.....	6
Software upgrades.....	6
Cueshell Quick Start	7
Opening a catalogue window.....	8
Copying or deleting an object	10
Copying, deleting or formatting a complete medium.....	11
Actions on a catalogue item.....	11
Generating a configuration file, generating a configured program copy and other configurations	13
Run time configuration	13
Startup configuration	14
Loading a configuration.....	15
The main menu items	16
Cueshell topics in alphabetical order.....	20
The Pointer Environment	32
The Pointer Interface	32
Multitasking with the pointer interface	33
The Window Manager.....	34
The Hotkey System.....	36
SuperBASIC extensions in the Hotkey system	36
Extra parameters:.....	37
Hotkey System SuperBASIC commands	38
Cueshell keyboard selection keys	43

Introduction

Cueshell is a desktop program intended to perform some everyday tasks in the computer in an easy way. The pointer driven user interface offers a way of instant access which is not possible on a command line level. Thus controlling basic operations like copying files, starting programs, etc. becomes possible without previous knowledge of a command line language. Program control is mainly intended to be performed with a mouse. Actions are generally produced by moving the mouse pointer over an option offered by the program and pressing one of the mouse buttons. If you are not already familiar with the features of the pointer driven user interface, please read the Pointer Environment chapter for more detailed information.

The main control elements of a mouse driven program are the pointer and the mouse buttons:

The pointer sprite

The pointer sprite is a small graphical object which is normally intended to be moved over the screen with mouse. As the Pointer Environment allows keyboard control as well, the pointer sprite can be moved with the cursor keys too. The pointer sprite has normally the form of an arrow. The sprite may change its shape over a certain region of a window or after a special action, thus signalling that a special action will be generated or is actually performed.

HITs and DOs

A mouse normally has two or three buttons, where the left and right buttons have predefined functions, HIT and DO.

(On a three-button mouse the centre button normally generates ESC.)

HIT is generated when the left mouse button is pressed. As the Pointer Environment allows keyboard control as well, pressing the SPACE bar also generates a HIT.

A HIT means that the related program option is selected, which is more than simply being available. A selected item is normally marked in a highlighted colour or with a graphical object in an obviously different shape. A HIT normally does not lead to an action, only when a pure selection obviously does make no sense or if the related action is easily reversible.

DO is generated when the right mouse button is pressed. As the Pointer Environment allows keyboard control as well, pressing ENTER on the keyboard generates a DO.

A DO means that the related program action is performed. This is normally not easily reversible, therefore before pressing DO you should be really sure about what will happen. A good program will always offer a chance, e.g. by presenting a special dialog menu, to cancel an absolutely irreversible action.

Sometimes an item may have several stages of being selected, or several actions may be coupled with it. In those cases HIT and DO are often used to generate alternatives, where HIT normally generates the weaker alternative.

Sometimes more than two alternatives may be required, then keeping a button pressed may lead to another action than releasing it immediately.

Keyboard control

As the pointer Environment allows keyboard control as well, but moving the pointer with the cursor keys is fairly inaccurate, most items normally do have a keyboard selection key where possible. Pressing the selection key normally is similar to moving the pointer over the item and pressing HIT. Where not obvious, the key is either marked explicitly or by underlining a character of the item description. Unfortunately in lists generated at run time it is not easy to mark a predefined selection key. Also when graphical objects are used to mark an item, the keyboard selection may not be obvious. As with our SERMouse driver it is very easy today to get a mouse on a standard QL to work, we have not made special efforts to make keyboard control perfect. Where necessary, please refer to the

-> list of keyboard controls at the end of the manual.

License Agreement

As a user of the Cueshell Software you agree with the following terms:

The Cueshell program is copyrighted by Albin Hessler Software, which remains the owner of this software. The user is granted the non-transferable license to use this software on a single computer. The user may make safety and working copies for his own purposes. The user accepts, not to give away or sell any copy of the software. The manual is property of the user but is also copyrighted by Albin Hessler Software and may not be copied.

This license becomes only valid if the filled in user registration form is sent back to Albin Hessler Software.

Limited warranty

The Cueshell program and this manual have been developed with great care. As we can not control all possible hard- and software constellations under which the program may be used, we can not give any warranty for the perfect functioning of this software at any time. Especially we are not responsible for resulting damages of a malfunction which may occur by any reason in conjunction with the usage of this software.

However we warrant for an error free diskette with an error free copy of the program. In case of any damage which occurs under normal use, we replace the disk without charge within a period of six months, provided the user registration has been sent back to Albin Hessler Software.

Software updates

Software updates are normally free if the original disk is sent to us together with the cost for the return postage, which can be covered by international reply coupons.

Software upgrades

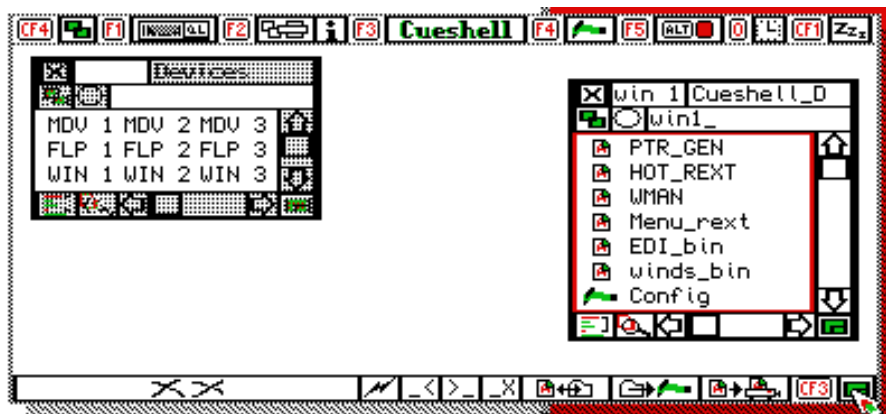
In case of major changes to the program we are free to offer special upgrade versions to all registered users for a special price.

Cueshell Quick Start

- start Cueshell as a job:
load the Pointer Environment and start Cueshell with EXEC or EX
e.g. insert the disk in FLP1_ and type **EX FLP1_Cueshell**
- load the resident version CUE_REXT:
load the Pointer Environment and then CUE_REXT
e.g. insert the disk in FLP1_ and type **LRESPR FLP1_Cue_rext**
- after CUE_REXT was loaded residently or if Cueshell is part of SMSQ/E as a module, type **CUE**

After the load, the Cueshell main window appears:
From left to right:

Item top	Key	Items bottom	key
move window	CTRL F4	Delete	D
QDOS-System control	F1	one level up	<
catalogue windows list	F2	one level down	>
Cueshell program options	F3	ending	X
job list	F4	DATA default	D
hotkey list	F5	PROGram default	P
date/time	0	printer DESTination default	SHIFT F4
sleep	CTRL F1	change size	CTRL F3



In the catalogue window area, the devices window is always present. Invisible parts of the device list can be reached by scrolling.
Change the size and position of the main window as required. The main window is limited to the minimum size as seen above and may cover the whole screen in any resolution. Use the configuration to set a size and position for startup.
The position and size of the devices window can be changed within the Cueshell window.

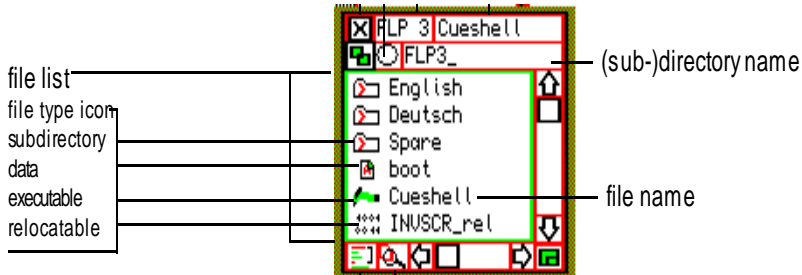
A DO on the move item moves it to the top left corner, a DO on the change size item brings it to maximum size at once.

Opening a catalogue window

A DO on a device name will open a catalogue window:

Catalogue window control and informations

No item	key
jump into the list	TAB
top of the list	SHIFT ↑
	SHIFT ↓



top row	key
close	ESC
device name and number	
medium name / refresh	M or R

second row	key
move	CTRL SHIFT F4
select all	A
path name	

right side	mouse	key
top arrow	HIT: scroll list one line up DO: scroll list one page up	ALT ↑ ALT SHIFT ↑
bar	move list to selected position continuous scroll	click bar at position drag
bottom arrow	HIT: scroll list one line down DO: scroll list one page down	ALT ↓ ALT SHIFT ↓

bottom row	mouse	key
sort		S
view		V
left arrow	HIT: pan one column left DO: pan one page left	ALT ← ALT SHIFT ←
bar	move list to selected position continuous scroll	click bar at position drag
right arrow	HIT: pan one column right DO: pan one page right	ALT → ALT SHIFT →
change size		CTRL SHIFT F3

Initially a catalogue window appears in the default size and sort order, which is configurable. Resize and position the window according to your needs.

Select **Size/sort order** **Save** from the **Cueshell** menu

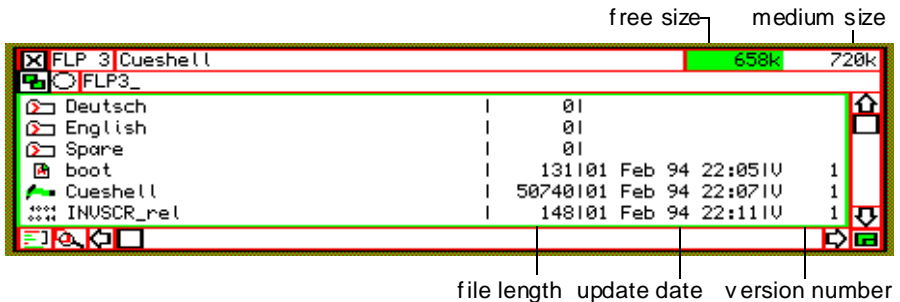
to save the actual size and position of the window onto the medium. Next time the directory is opened, the catalogue window will appear in the saved size if possible.

The catalogue windows are very flexible. They wander and shrink according to their position and the main window size, but they never forget their original size and position, unless these are set explicitly. All catalogue windows are always drawn, even if their position would be outside the actual main window. They then shrink to the minimum size in either direction. If windows are covered totally, they can be picked on top using the window list (F2) menu.

A DO on the move item, will move the window to the top left corner of the catalogue window area.

A DO on the change size item will pop up the window to the actual maximal possible size.

Now the additional information can be seen:



- the actual free size on the medium
 - the total medium size
- and in the list:
- the file length
 - the update date
 - the version number

Please notice that Cueshell, if possible, always preserves the additional file data when copying a file, i.e. especially the update date is treated as a creation date and preserved.

Copying or deleting an object

Cueshell has for copy modes:

- update
- move
- backup
- delete

Cueshell always copies **objects** which may be a single file, a complete directory or a complete medium.

Select one or more files in the catalogue window by HITing either the file type item or the file name item beyond the name.

You must see the normal arrow sprite to select:

DO on the last file to select or onto an already selected file, then an animated sprite appears, the copy sprite which has 5 shapes:



Move the pointer over the destination and HIT. The destination can be:

- a device name in the devices list
- a catalogue window
- a sub-directory item in a catalogue window
- the delete bar

To determine the destination it is sufficient to see and point to a small part of it.

When the destination is selected, a dialog window appears which asks to confirm the action. If the destination device is different from the source, update or backup mode can be chosen:

In **update mode** only those **files** are copied which already exist on the destination.

In **backup mode** the source **object**

(which might be a file, a sub-directory or a complete medium) is completely copied to the destination. If there is

already an object with the same name, it is overwritten. This means that an existing object on the destination is deleted and completely replaced by the source object.

For sub-directories or complete mediums this means that all the content is deleted (which might be a whole tree).

If the destination device is the same as

the source, the **move option** is offered instead of the update option. Moving a file

simply means a rename on the medium, i.e. only the name in the file header is

changed and the file itself is kept unchanged.



If the delete bar is selected as destination, the delete box asks to confirm the delete action.



Copying, deleting or formatting a complete medium

Selecting a complete medium as source and/or target in the devices window is a little bit different from the method to select a file or sub-directory in a list.

A HIT on a device enters copy mode (the animated copy sprite appears). A DO on another device initiates copying. Update or backup mode can be chosen as options.

If the delete bar is HIT, then it can be chosen to delete the complete content of this medium, or to format it. The device name is shown. If there is a medium found, the old medium name is presented.

You can type a new name or any other format option. Please notice that some devices (for example RAM disk) return a medium name which is not a valid format string option, so it must be stripped away.



Actions on a catalogue item

There are two possible ways to generate an action on the **file type item**:

- **HIT** only selects the file. The selected file will be part of a subsequent action which operates on files
- **DO** enters copy mode. To unset (cancel) copy mode, HIT on a free space outside the catalogue or devices windows, or select any loose item.

There are several possible ways to generate an action on the **name item**:

- **HIT beyond the name** only selects the item for further actions
- **DO beyond the name** open the view window and presents the file

When the pointer is *over the name of the file*, the I-bar appears.



This means that any following action is coupled with the file directly.

- **HIT on the name** stuffs the name into the **Hotkey buffer**. The item is not selected then! If you're not quite sure, whether the name was stuffed or not, just do it again. The Hotkey System does never stuff the same string twice!
- **HIT on the name and the button kept pressed** enters **edit mode**. When the text cursor blinks, the name can be edited. ENTER confirms to rename.
- **DO on the name** generates an action which depends from the file type:
 - sub-directory file open sub-directory
 - other files when FileInfol is present, it is called first. If it takes over control, Cueshell will not do anything more.
 - data file no action on a single DO (see below)
 - executable file execute file
 - relocatable file no action

- **DO on the name and the button kept pressed** enables additional options:
 - data file open sub-menu to stuff SuperBASIC command. Super-BASIC is picked on top and the command together with the file name is stuffed into the channel #0 keyboard queue
 - executable file open sub-menu to select PSION, GUARDIAN or to edit a command string
 - sub-directory close actual catalogue window and read sub-directory to new window

Generating a configuration file, generating a configured program copy and other configurations

Configuration happens in two steps:

1. Configuring the Cueshell directory with Config or MenuConfig.
When Cueshell runs as an executable job from file, the Cueshell directory can also be set together with a new configured program copy. There are menu items for this in the options menu.
2. From within the program, by saving the actual program status.



Run time configuration

is possible for the colours and the Cueshell Hotkeys.

This can be done from within the Configuration menu accessible through the Cueshell (F3) menu.

Type new Hotkeys to pick or use Cueshell and then DO the Set now item.

Select the Colours item to open the window for colour configuration.

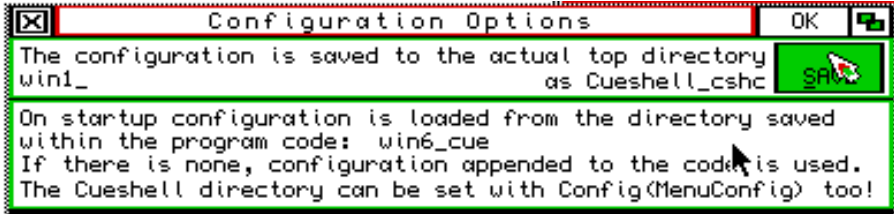
Also some system parameters can be altered at run time. Use the keyboard and mouse menus accessible through the system menu (F1).

Startup configuration

is made by selecting the **Save** item in the configuration menu. This saves the configuration in a file `Cueshell_cshc` into the directory which is actually the top catalogue window (if there is no window open other than the devices window, the configuration is cancelled). This directory is not automatically opened at startup, unless the appropriate option was selected in the configuration menu. The following parameters are saved:

- actual main window size
- actual main window position
- actual Cueshell Hotkeys
- actual colours
- keyboard parameters (when the option is selected)
- mouse parameters (when the option is selected)
- the last set sleep position
- catalogue window default size and sort order. These are taken as set for the actual top window
- actual DATA, PROG and DEST default (only when the option **Set path defaults** is selected)
- all names of the actually opened directories

After selecting the **Save** item, an additional options menu appears:



Loading a configuration

On startup Cueshell will try to (in this order):

- open a file Cueshell_cshc which was passed as a command string and to use this configuration.
- search for Cueshell_cshc in the Cueshell directory as set with Config or MenuConfig.
- otherwise it will use the configuration saved with the program code.
! A new copy of the program code can only be generated from within a job which was started with EX. It is not possible from within a Cueshell started from the resident module (Cue_rext or SMSQ/E module)!

From the configuration file, the above mentioned sizes and options are taken and all directories saved with the configuration are opened. Errors will be ignored, i.e. directories not present will simply be ignored.

Object related configurations are possible on directories and single files:

The position, size and sort order of any directory can be saved (and deleted) at any time using the appropriate option in the Cueshell (F3) menu. Changes to an open catalogue window do not alter the saved parameters.

Single files can be declared to be write protected and/or invisible. This is performed using the appropriate option in the Cueshell (F3) menu.

The main menu items

- from left to right
- please read the alphabetical section for more detailed information on special tasks

Move

CTRL F4

Press HIT to move the main window. The move window sprite appears. Move to new position and press HIT again.



System control

Select sub.menu to set

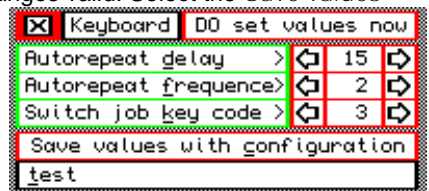
- keyboard parameters
- mouse parameters
- clock

F1



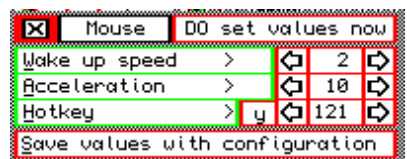
Keyboard

Set values by using the left/right arrows or select the number window to type a new value. Select DO set values now to make changes valid. Select the Save values with configuration option, if you want the actual values to be saved with a configured program version. The effect of changes to the keyboard parameters can be seen by typing to the test-window.



Mouse

Set values by using the left/right arrows or select the number window to type a new value. Select DO set values now to make changes valid. Select the Save values with configuration option, if you want the actual values to be saved with a configured program version. *The mouse Hotkey is generated by the pointer-interface when the left and the right mouse button are pressed simultaneously.*

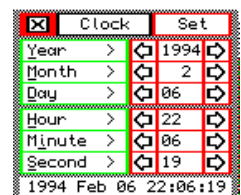


Clock

Set values by using the left/right arrows or select the number window to type a new value. Select Set to make changes valid. Move the pointer over the date/time item at the bottom line to see the clock run. This also updates the values in the number items.

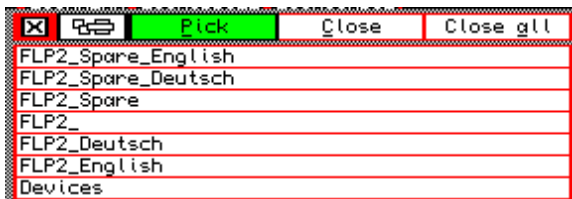
Clock

Set values by using the left/right arrows or select the number window to type a new value. Select Set to make changes valid. Move the pointer over the date/time item at the bottom line to see the clock run. This also updates the values in the number items.



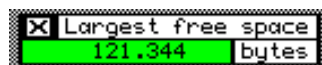
Windows control F2

Cueshell catalogue windows can either be picked on top by HITting them directly or via the windows list, especially if a window is completely covered by others. The list can also be used to close one or all windows. Select the option (Pick is preselected) then DO the name.



Info

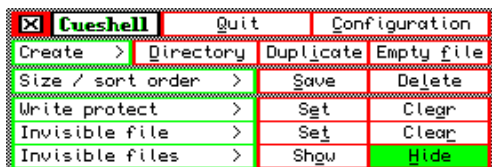
Shows the amount of largest free space in memory. HIT anywhere or move the pointer outside to remove the information.



Cueshell

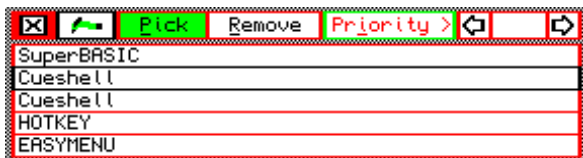
The Cueshell command menu offers options to

- Quit Cueshell
 - Create a new directory
 - Create an empty file
 - Duplicate a file
 - Save or delete the size and sort order of the top catalogue window into the directory on the medium
 - Set or clear file attributes (hidden and write protected)
 - Select option to show or hide invisible files
 - Open the configuration menu
- here the Cueshell hotkeys, the colours ... can be changed at run time.
See Generating a configuration file...



Job list

This sub-menu allows to pick on top, remove or change the priority of any actually running job. To pick or remove, select the option then DO the jobs name. To change the priority, HIT the jobs name. The actual priority is then shown. Use the arrows or type a new value into the number item directly. Finally DO the value item.



Hotkeys list F5

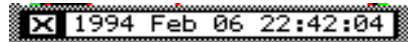
This sub-menu allows to DO Hotkeys (as if typed on the keyboard). This is especially useful if you've forgotten the key. You can also remove Hotkeys or switch them on and off. Select the option, then DO the name item.



Clock

0

Shows the actual system date and time. HIT anywhere or move pointer outside to quit.



Sleep

CTRL F1

Press HIT to enter Cueshell sleep mode -> Cueshell shrinks to main menu line.
Press DO to use the QPAC2 button frame.



Delete item

This item at the bottom of the main menu window must be selected to delete files.

Refresh catalogue window

CTRL F2

The directory content of the top window is updated

Wild card names and endings

To use these items, a name must be selected in the catalogue window:

- _< one step up to the next underscore
- _> one step down to the next underscore
- _X show only those files Which have the same ending as the selected file
- a DO on the selected item will toggle between the files having, and those not having the ending.
- HIT returns to the full list

DATA path

PROGram path

HIT opens the menu to edit or set the path to the actual directory or the default saved with the configuration.

DO opens the directory or picks it on top.

Printer DESTination

HIT on the item to open the menu to edit or set the device. The name can be edited or set to a selected file. Also the default can be reset.

When the FF item is selected:

- a form feed is sent to the printer at the end of a file
- if the destination is a file, the new file is appended.

Printing

DO on the printer item starts printing. If no file is selected, the directory is printed, otherwise all selected files are printed.

Change size CTRL F3

HIT the item. The change size sprite will appear. Position the sprite where you want the bottom-right corner of the main window, then HIT. If the position is below the minimum size, this will be set, otherwise any size up to the physical screen size can be set.



Cueshell topics in alphabetical order

Backup

One of the Cueshell copy modes.

A complete object is copied. All file header information is preserved. If an old object with the same name exists on the target device, it is overwritten, i.e. the old content is deleted and completely replaced by the source.

Please notice that an object can be a sub-directory or a complete disk! In those cases the complete content is deleted ruthlessly and replaced by the source object.

If (and only if) an object is a single file, the update date is checked and if the target is as old as the source or newer, Cueshell will ask to replace this file (and others) or not.

Catalogue window size and position

Catalogue window handling is very flexible within the catalogue window area. A window always knows two sizes and origin positions

- the size and origin position explicitly set
- the actual possible size and origin position

A window is always forced to be within the main window but never forgets its dream size and origin position, unless it is changed using the catalogue window move and/or change size items. -> Save size and sort order

Clock

HITing the clock item at the top right side of the main menu will open a sub-window which shows the actual date and time. The clock is running as long as the window is open. To close the window HIT anywhere or just move the pointer out of it.

-> Setting the clock.

Configuration

Select the Cueshell menu (F3) and then Configuration.

Configuration is always made from the actual program status. Thus this method allows generating a working copy of the program on a startup disk as well as saving the actual status for temporary use, for instance to continue work the next day from a known stage. Configurable items are:

- > colours
- > Cueshell Hotkeys
- > keyboard parameters
- > mouse parameters
- > program startup size
- > program startup origin
- > program sleep position
- > directories to be opened at startup
- > default catalogue window size and position
- > default catalogue window sort order
- > catalogue window size and position
- > catalogue window sort order
- > file access keys, write protect and invisible

Copy directory

DO on the file type item of the directory to activate copy mode. Then move the pointer over the target and HIT. Select backup or update (move) mode, then confirm to copy.

Copy disk

To copy a complete medium, HIT the name of the device in the devices window. This will enter copy mode immediately. Then DO on the target in the devices window. Select backup or update mode, then confirm to copy.

ATTENTION:

In backup mode all files on the target medium will be deleted ruthlessly. So use this option with great care and never on your hard disk!

Copy file

DO on the file type item of the file to activate copy mode. Then move the pointer over the target and HIT. Select backup or update (move) mode, then confirm to copy.

Copy mode

Devices window:

HIT the name of a device to enter copy mode. The animated copy sprite will appear.

Catalogue window

Select one or more files by HITing the file type item or beyond the characters of the name

DO the file type item of a file. The animated copy sprite will appear.

The copy action is initiated by HITing the target.

Possible targets are

- a device in the devices list (DO it if the source is a full device)
- a catalogue window
- a sub-directory in the file list of a catalogue window
- the delete bar at the bottom of the main window

Copy mode is unset by HITing outside any catalogue window or if any item action is selected.

Create directory

Select the Cueshell menu (F3), then DO the Create Directory item. An empty directory with the name New will be created on the top window directory. HIT the name and keep the button pressed until the cursor blinks to edit the name. Confirm with ENTER, then the new directory is renamed.

Create duplicate

Select the Cueshell menu (F3), then DO the Create Duplicate item. A duplicate of any selected file will be created on the top window directory. The duplicate will have the number 2 appended to the original name. HIT the name and keep the button pressed until the cursor blinks to edit the name. Confirm with ENTER, then the file is renamed.

Create empty file

Select the Cueshell menu (F3), then DO the Create Empty file item. An empty file with the name Empty will be created on the top window directory. HIT the name and keep the button pressed until the cursor blinks to edit the name. Confirm with ENTER, then the empty file is renamed.

Delete catalogue window size and sort order

Select Delete Size/sort order from the Cueshell menu (F3).

Delete file

DO on the file type item of the file to activate copy mode. Then move the pointer over the delete-bar and HIT. Confirm to delete.

Delete directory

DO on the file type item of the directory to activate copy mode. Then move the pointer over the delete bar and HIT. Confirm to delete. To empty a directory, open it, select the all item and DO on the file type item of one file in the list to activate copy mode. Then move the pointer over the delete item and HIT. Confirm to delete.

DEV device

Cueshell supports the SMSQ DEV device. The DEVx is replaced by the original path. This also works with wild cards.

Edit line

Cueshell has a special line editor, which accepts all standard editing commands:

left/right	move one character left/right
SHIFT left/right	jump one word left/right
ALT left/right	jump start/end
ALT SHIFT left/right	PAN left/right
CTRL left/right	delete character left/right
CTRL SHIFT left/right	delete word left/right
CTRL ALT left/right	delete line/delete to end of line

Where the editable string length is longer than the window width, the text can be PANed.

Editing is normally starts where the insert pointer is shown. Editing can be started by HITing any character of the string. To enter editing, the mouse must be kept still (see below). By this, as it is intended to type from the keyboard anyway, it is normally better to position the insert point using the mouse and then entering editing by pressing SPACE instead of the mouse button.

The line editor can be left with Cursor up/down, ESC, ENTER or simply moving the mouse. It depends from the special situation which action accepts or cancels editing.

Normally ENTER is treated to be OK and the new string is taken, while the others are treated as ESC and the old string is preserved.

Execute job

DO on the filename of a name item of an executable file. For more options -> Start program

Execute PSION program

DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of a PSION program and keep the button pressed until the PSION/GUARDIAN sub-menu appears. DO the PSION item. A 512x256 pixel guardian window will be opened and you will be asked to enter a memory size for the program. This is to prevent the PSION program from grabbing all the available free memory. Thus multi-tasking the PSION programs is possible.

Execute job which needs a guardian window

Some older programs which write directly into the screen memory (bypassing the screen driver) need to be executed with the GUARDIAN option.

DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of the program and keep the button pressed until the PSION/GUARDIAN sub-menu appears. DO the GUARDIAN item. A 512x256 pixel guardian window will be opened for the program.

If more options are needed to make badly behaving software run with the Pointer Environment please use the parameters offered with the SuperBASIC commands in the Hotkey system.

Execute job with command string

If a command string shall be passed to the job, DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of the program and keep the button pressed until the PSION/GUARDIAN sub-menu appears. Type the command string and press OK.

FileInfo

Cueshell supports the FileInfo Thing, i.e. on a DO over the name of a file always the FileInfo Thing is called first. For more information please read the FileInfo instructions.

Hotkeys

Cueshell always starts with two predefined Hotkeys. The keys can be altered to match your individual preferences or needs from within the Cueshell configuration. The Hotkeys will be removed if you leave Cueshell from within the program. If Cueshell is removed externally, the Hotkeys will still be present, as they belong to the system. It is easy to remove them with the SuperBASIC function HOT_REMV supplied by the Hotkey system, although they will not do any harm.

Pick Cueshell

Initially set to ALT Y

This Hotkey is used to pick Cueshell on top. If, by any reason, you have several Cueshell jobs running that all have the same pick Hotkey, then the Hotkey system picks them in turn.

Use Cueshell

Initially set to ALT X

This Hotkey is used to pick and use Cueshell from any program. Cueshell will be picked on top but knows from which job it was picked. Now, if you type ESC within Cueshell, the calling job is picked on top again and:

- the name of the active directory or a selected file within it is stuffed into the calling jobs keyboard queue
- put the Cueshell arrow outside any catalogue window if no string shall be stuffed

This procedure only works from one calling job at a time. If you've once picked Cueshell with this Hotkey without returning to the calling job, then the first time you try to use it again from another job, the Hotkey system will sound, thus reporting it is unable to use this key. But on the second attempt, the previously calling job is forgotten and it works again.

To let the Hotkey simply forget the calling job, type it from within Cueshell.

As Cueshell must wait to stuff the name until the target window is drawn and its keyboard queue is active, Cueshell is suspended for a short time.

Programmers hint: Use HOT_DO (SuperBASIC) or HK.DO (Assembler), if you want to pick or pick and use Cueshell from within your program directly.

Load SuperBASIC extensions

DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of a SuperBASIC extension file and keep the button pressed until the LRESPR/QLRUN/LRUN sub-menu appears. DO the LRESPR item. SuperBASIC will be picked on top and the command together with the file name is stuffed into the command line.

This only works if SuperBASIC is ready to wait for command line input!

Attention! If the file is not a SuperBASIC extension file, your system will crash!

Memory information

The largest free space in the transient program area is shown if you HIT the information item.

To close the window HIT anywhere or just move the pointer out of it.

Move

One of the Cueshell copy modes.

If source and target of a copy action are on the same medium, Cueshell offers this mode instead of the update mode; as the update mode is mainly intended for safety copies it makes no sense on the same medium. In practice moving a file on the medium simply means a rename.

New Directory

Select the Cueshell menu (F3), then DO the Create Directory item. An empty directory with the name **New** will be created on the top window directory. HIT the name and keep the button pressed until the cursor blinks to edit the name. Confirm with ENTER, then the new directory is renamed.

Object

The Cueshell file manager is object orientated. An object may be:

- a single file
- a sub-directory tree
- a complete medium (e.g. a floppy disk)

Open a catalogue window

DO the name of a device in the devices window.

DO the name of a sub-directory file in the file list of a catalogue window.

Pick a catalogue window on top

A HIT somewhere in the area of an inactive window will pick it on top.

A window can also be picked on top using the catalogue windows list accessible through the windows item (F2). This is particularly useful if a window is mostly or complete covered by another window.

The catalogue windows list can also be used to close a window without picking it on top, or to close all windows at once.

Print

A DO on the printer item starts printing to the selected device (HIT the item to set the printer destination).

- the actual directory is printed out, when no file is selected in the list
- other the files are printed

PSION program

-> start PSION program

Quit Cueshell

Select the Cueshell menu (F3) and then Quit.

Rename a file

HIT the position in the file name where you want to start editing and keep the button pressed until the text cursor appears. As you intend to type from the keyboard, you may prefer to use SPACE instead of the left mouse button, otherwise you must keep the mouse still (see below).

Editing is also possible if the name is longer than the window width.

Moving the mouse, ESC or Cursor key up/down will cancel editing. Type ENTER to confirm the new name. The file will then be renamed.

Non-empty sub-directory files can not be renamed to a longer name.

Unfortunately the level2 device drivers do not reset emptied sub-directory files to zero length. Therefore Cueshell scans the sub-directory file before entering edit mode.

This is to protect you against destroying your sub-directory tree, as somewhere down in the tree the total path name might get too long leading to bad name errors.

Rename a sub-directory

-> rename a file

Empty sub-directories can be renamed like a file.

Non-empty sub-directories can only be renamed directly to a name with the same or lesser length.

To rename a non-empty sub-directory to a longer name, create a new directory and move the content in it.

Save catalogue window size, position and sort order

For any (sub-)directory the catalogue window size, position and sort order can be saved onto the medium.

Open the directory, then select the desired size and position using the move and change size item.

Select the sort item in the bottom left corner. Here up to four different sort criteria can be selected

- | | | | |
|---|------|-----------------|---------------------|
| - | name | the file name | alphabetically |
| - | time | the update date | newest on top |
| - | type | the file type | sub-directory files |
| | | | data files |
| | | | executable files |
| | | | relocatable files |
| - | size | file length | smallest on top |

Cueshell sorts the strongest criteria last, i.e. if for instance you want the list to be sorted by file type and within file types by name, select name in the first row and type in the second row. Do not select sort steps unnecessarily as any step takes some time, the more the longer the directory is.

Finally select Save Size/sort order from the Cueshell menu (F3).

Setting the Clock

Select Clock from the system control menu (F1). Set the date and time using the left/right arrow items or type new values directly. DO the Set item. The actual system date and time are shown at the bottom of the sub-menu window. Move the pointer over this item to make the clock run.

Sleep Cueshell

Cueshell has two sleep modes.

1. HIT the sleep item in the top right corner, and the Cueshell menu window will shrink to the top menu line. Position the window in sleep mode where you like it. This position is saved, and the next time the sleep item is HIT, it will automatically move to that position. The selected sleep position is also saved with the configuration.
- HIT any item in sleep mode to wake Cueshell. HITing the sleep item will simply redraw the main window, while HITing another item will redraw the main window and enter that option at once.
2. DO the sleep item and Cueshell will use the QPAC2 button_sleep thing if present.

Sort a directory list

Select the sort item in the bottom left corner of a catalogue window. The sort sub-menu appears.

Here up to four different sort criteria can be selected

- name the file name alphabetically
- time the update date newest on top
- type the file type sub-directory files
 data files
 executable files
 relocatable files
- size smallest on top

Cueshell sorts the strongest criteria last, i.e. if for instance you want the list to be sorted by file type and within file types by name, select name in the first row and type in the second row. Do not select sort steps unnecessarily as any step takes some time, the more the longer the directory is.

Cueshell always saves the *natural* order in which the files are stored in the directory. So any sort order can be cleared without re-reading the directory from the medium.

Select Sort to initiate sorting.

Start program

- **executable file**

DO the name item of an executable file.

- **PSION program**

DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of a PSION program and keep the button pressed until the PSION/GUARDIAN sub-menu appears. DO the PSION item. A 512x256 pixel guardian window will be opened and you will be asked to enter a memory size for the program. This is to prevent the PSION program from grabbing all the available free memory. Thus multi-tasking the PSION programs is possible.

- **executable program which needs a guardian window**

some older programs which write directly into the screen memory (bypassing the screen driver) need to be executed with the GUARDIAN option.

DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of the program and keep the button pressed until the PSION/GUARDIAN sub-menu appears. DO the GUARDIAN item. A 512x256 pixel guardian window will be opened for the program.

If more options are needed to make badly behaving software run with the Pointer Environment please use the parameters offered with the SuperBASIC commands in the Hotkey system.

- **executable program with command string**

if a command string shall be passed to the job, DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of the program and keep the button pressed until the PSION/GUARDIAN sub-menu appears. Type the command string and press OK.

- **SBASIC programs**

are shown by an own icon, when the ending is _bas. These programs are treated like executable jobs (SMSQ only)

- **SuperBASIC program**

DO the characters of the name (the pointer must be over the characters of the name and must have changed to the insert pointer) of a SuperBASIC program file or Qliberator _sav file and keep the button pressed until the LRESPR/QLRUN/LRUN sub-menu appears.

DO the LRUN item (or QLRUN for a Qliberator _sav file). SuperBASIC will be picked on top and the command together with the file name is stuffed into the command line.

This only works if SuperBASIC is ready to wait for command line input!

Attention!

If the file is not a SuperBASIC program file, the system may crash!

Stuff file name into Hotkey buffer

HIT on the characters of the name. The pointer changes to the insert pointer when it is over the name. The item will not be selected then! If you're not quite sure, just do it again. The Hotkey system never stuffs the same string twice!

Update

One of the Cueshell copy modes.

All files of the object are only copied if a file with the same name already exists on the target. All file header information is preserved.

The update date of every file is checked and if the target is as old as the source or newer, Cueshell will ask whether to replace this file (and others) or not.

View a file

Select one or more files in the directory list, then select the view file item at the bottom of the catalogue window (the magnification-glass icon) or DO a file name item beyond the name.

The catalogues will disappear and the view file window will be shown instead. The first selected file will be shown.

HIT the up or down arrow at the right side to scroll up or down a line.

DO on the arrows to scroll up or down a page.

HIT on the slider bar, to position the file pointer accordingly.

All 255 characters of the character set are shown. Control characters below \$20 and beyond \$C0 are represented by a character giving their Hex value.

Line feed (LF), carriage return (CR) and carriage return/line feed(CRLF) are represented by special characters. All three generate a single line feed in the view window. Otherwise a line feed is forced when the window line is full.

There are three view modes, which can be toggled by HITing the item in the top right corner.

HIT the right/left arrow to view the next/previous selected file.

Scrolling forth and back and opening the next/previous selected file is also possible with the arrow pointers which appear directly over the view window.

Wake Cueshell

HIT any item in sleep mode to wake Cueshell. HITing the sleep item will simply redraw the main window, while HITing another item will redraw the main window and enter that option at once.

The Pointer Environment

The pointer environment is a system extension to QDOS. Since it appeared first with QRAM it has been further developed and enhanced. Therefore you should only use a recent version. The pointer environment is already built in SMSQ/E.

The Pointer Environment consists of three parts:

Pointer Interface	file 'ptr_gen'
Window Manager	file 'wman'
Hotkey System	file 'hot_rext'

The files do not come with Cueshell. All three parts must be loaded residently after the computer is switched on in the following order. If Toolkit 2 and Lighting are used, they must be initialized before:

```
100 _IngINIT
110 TK2_EXT
120 LRESPR 'flp1_ptr_gen'
130 LRESPR 'flp1_wman'
140 LRESPR 'flp1_hot_rext'
```

The Pointer Interface

The Pointer Interface is a system extension to QDOS. It installs the pointer device which makes it possible to move a pointer on the screen. Generally this is a small item (sprite) of max. 64 x 48 pixels. This sprite may be moved with a mouse or the cursor keys as well. Thus for instance

- a menu for program control may be managed
- a pencil in a paint program may be moved
- objects may be marked and moved around etc. ...

Although mainly intended for the use with a mouse, the pointer interface allows all control to be done with the keyboard alone. But a mouse gives more flexibility and comfort. For the standard QL the QIMI Mouse Interface as well as our SERMouse serial mouse driver are available. The mouse may even simulate the cursor keys when moved while the left button is pressed. Thus e.g. it may be used in any Editor (ED, QUILL etc.).

The Pointer Interface includes two SuperBASIC extensions CKEYON and CKEYOFF:

CKEYON

The mouse pointer can be moved with the cursor keys.

CKEYOFF

The mouse pointer can not be moved with the cursor keys.

Multitasking with the pointer interface

QDOS allows several programs to run at the same time (multi-tasking). But standard QDOS has no option to save the window content of a program (destructable windows). Each program must have a separate option to restore its windows. CTRL C is used to switch around the keyboard queues, but the window content is not restored and the program on top destroys the windows of other programs.

The pointer interface - by help of an extended channel definition - saves and restores the window content automatically when the programs are switched around. This extended channel definition is the reason why software which does not use the system routines correctly (e.g. by manipulating the channel definition blocks directly) does not run under the pointer environment. The Hotkey System and Cueshell offer several ways to adapt badly behaving software (especially the PSION suite). But all serious software suppliers now offer program versions that run with the pointer environment.

Programs (jobs) are switched around with CTRL C as with standard QDOS, but now the pointer interface switches around the pile of primary windows that belong to every job and no longer the simple keyboard queue.

The Pointer Interface distinguishes between two different types of windows of a job.

1. The primary window is the first window channel opened for this job.
2. All subsequently opened window channels are called secondary windows. The channel definition does not only keep the window size but also the window outline which is equal or beyond the window size area. Pointer requests are restricted to the outline area while all standard input/output is restricted to the window size area.

The primary window outline may cover the whole screen, while a secondary outline is restricted to be within its primary outline.

To keep the new window definition compatible with standard QDOS, window channels are managed by the Pointer Interface in two different ways.

With unmanaged windows (in general windows of those programs which are not written for the Pointer Interface) it watches automatically that the primaries outline always covers all window channel areas opened for this program.

When a primary window is marked to be managed (also called well behaved) by setting the outline explicitly, then the program itself has to make sure that all its windows are within its primary windows outline. Pointer requests are only possible on managed windows. All managed windows of a program (job) form a hierarchical structure. The Pointer Interface allows pointer requests only to the latest window declared to be managed.

Programmers access

In the pointer interface there are some system routines implemented as

TRAP #3 \$6C-\$7F

to which the assembler programmer has access. The technical description is in the QPTR manual.

Our EASYPTR development system comes with a set of SuperBASIC extensions which give access to the system routines of the pointer interface for the Super-BASIC programmer. For the C-programmer a set of library routines is available.

The Window Manager

The Window Manager is a set of vectored routines to manage the internal structure of a menu-window. As these can be used by several programs they are loaded as a system extension together with the pointer interface.

Structure elements

The internal structure of a menu window is built by several elements:

Main window

This is the total area within the outline. This area is defined by size, border, shadow and pointer sprite.

Information sub-windows

These can be situated at any place within the main window. They can be used for general information purposes, division of the window, colour design,

Info-windows are ignored by the pointer request.

Loose menu items

These are special areas for program control. When the pointer is within a loose item, a border is drawn around the item. Any loose menu item is directly coupled with an action routine, which is processed automatically when the item is selected (-> HITs and DOs).

Application sub-windows

For free use by the application program. Application sub-windows are, like loose items, subject of the pointer request, i.e. any application sub-window has a hit routine, where the application program can decide for its own purposes what to do, if the pointer is within the window.

Application sub-windows may have a regular menu structure also processed by the window manager. This menu may be larger than the window size. Hidden parts are scrolled or panned into the visible area by help of pan and/or scroll bars and arrows. The menu may consist of several independent sections and/or be able to be split in several sections (and joined again).

Control elements/Events

The window manager offers standardized elements to control basic functions of a menu.

key	event
SPACE	select
HIT (left mouse key)	select
ENTER	do action
DO (right mouse key)	do action
ESC	cancel
F1	help
CTRL F1	sleep
CTRL F2	wake
CTRL F3	change window size
CTRL F4	move window

Programmers access

The technical description of the window manager routines can be found in the QPTR manual.

Our EASYPTR development system offers an easy way to design menu-windows and structures on the screen and to control them from within SuperBASIC as well as C- and assembler written programs.

The Hotkey System

The Hotkey System is intended to connect various actions (starting jobs, inserting strings, picking jobs,...) to a single key press (together with the ALT key). Some key presses do have a standard meaning (although configurable):

ALT ENTER	recovers the last string stuffed into keyboard queue
ALT SPACE	recovers the last string stuffed into the hotkey buffer
ALT SHIFT SPACE	recovers the previous string stuffed into the hotkey buffer

The hotkey buffer can be used by any program, e.g. a filename can be written into the buffer by one program and inserted into the keyboard queue of an other program.

Beyond these, up to totally 128 Hotkeys can be defined.

A Hotkey may lead to one of the following actions:

- a string is inserted into the current keyboard queue as if it where typed on the keyboard directly
- a string is stuffed into the hotkey buffer
- an executable program is started
- an existing job is picked to the top
- a sleeping program is wakened
- SuperBASIC is picked on top and a command sequence inserted into the command line
- an executable program is loaded and marked to be an executable Thing for later use
- an executable Thing is started.

SuperBASIC extensions in the Hotkey system

Some SuperBASIC extensions to use and define Hotkeys are part of the system. Most of them are functions and are usually used in a *boot* program. If necessary they can be used directly from the command line.

A description of the commands follows. The syntax description follows the usual conventions. Alternative parameters are put between curly brackets, optional parameters between square brackets.

Extra parameters:

These parameters can be set where **[,extra[,...]]** is to be found in the description.

Their purpose is to adapt difficult programs.

P	ask for memory size
P,kByte	the program is given the memory size given in kByte

The P option is mainly intended to be used with the PSION suite programs, which otherwise would grab all available memory.

Example: EXEP 'flp1_quill',P,64

I	the program is marked to be impure (not romable).
----------	---

Example: ERT HOT_RES ('c','flp2_c1mon',I)

U	the primary window of the specified program is marked to be unlockable, that means it will destroy other windows as in standard QDOS.
----------	---

Example ERT HOT_LOAD1 ('s','flp2_mysuperwatch',U)

G	opens a guardian window (primary window) for this program, which covers the whole screen.
----------	---

G,width,height,x-origin,y-origin	opens a guardian window (primary window) in the given size and position.
---	--

The G parameter is mainly intended for programs which do not use the standard screen input/output routines, e.g. if the program writes directly into the screen buffer.

An additional command string parameter, as with EX in Toolkit 2, can be given with the functions HOT_RES, HOT_CHP, HOT_LOAD and HOT_THING.

Example ERT HOT_RES ('I','flp1_linker';'-with flp2_link')

Hotkey System SuperBASIC commands

ERT

ERT

Automatic error report procedure to be used with functions. If the function return value is set to the error code, ERT will redirect it to the command channel.

EXEP

EXEP {filename}{thing [, extra[,...]]

The job or thing is started.

HOT_CHP

err = HOT_CHP (key\$, filename [, extra[, ...]])

The file is loaded into the common heap area. A new copy of the program is started every time when ALT+ key\$ is pressed.

Example: ERT HOT_CHP('q', 'flp2_QD')

Variant:

err = HOT_CHP1 (key\$, filename [, extra[, ...]])

Only one copy of the program can be started. (For not romable programs.)

HOT_CMD

err = HOT_CMD (key\$, commandline\$)

Picks SuperBASIC and stuffs commandline\$ into the SuperBASIC command channel #0.

Example: ERT HOT_CMD ('F','print free_mem',")

HOT_DO

HOT_DO {key\$}{name\$}

Executes the action defined for this Hotkey (key\$) or Thing (name) linked to this Hotkey, as if it was typed on the keyboard.

HOT_GO

HOT_GO

Starts the hotkey job. Then all defined hotkeys are active.

HOT_KEY

err = HOT_KEY (key\$, string\$[, string\$[, ...]])

The string(s) is(are) stuffed into the keyboard queue when the Hotkey is pressed.
An empty string will cause a line feed

Example: ERT HOT_KEY ('Y', 'Yours sincerely',")

HOT_LIST

HOT_LIST [#channel]

Gives a list of all actually defined Hotkeys on the given channel (default: #1).

HOT_LOAD

err = HOT_LOAD (key\$, filename [, extra[, ...]])

A new copy of the program is loaded and started. Then, every time when ALT+key\$ is pressed a new copy is started with the same code.

Example: ERT HOT_LOAD ('s', 'flp2_easysprite_exe')

Variant:

err = HOT_LOAD1 (key\$, filename [, extra[, ...]])

Only one copy of the program can be started.

HOT_NAMES\$

name\$ = HOT_NAMES\$ (key\$)

Returns the name of a hotkey.

HOT_OFF

err = HOT_OFF ({ key\$ } { name })

The hotkey is switched off (not removed).

HOT_PICK

err = HOT_PICK (key\$, jobname)

The job is picked to the top.

HOT_REMV

err = HOT_REMV ({ key \$}{ name })

Removes a hotkey.

HOT_RES

err = HOT_RES (key\$, filename[, extra[, ...]])

The file is loaded into the resident procedure area. A new copy of the program is started every time when ALT+ key\$ is pressed.

Example: ERT HOT_RES ('s', 'flp2_easysprite_exe')

Variant:

err = HOT_RES1 (key\$, filename[, extra[,...]])

Only one copy of the program can be started. (For not romable programs .)

HOT_SET

err = HOT_SET ([newkey\$,]{key\$}{name})

Reset a hotkey previously switched off with HOT_OFF. Additionally a new key may be defined.

HOT_STOP

HOT_STOP

Removes the hotkey job. Then all defined hotkeys are inactive. The hotkey job can be removed via the Jobs list in Cueshell or RJOB of TK2 too.

HOT_STUFF

HOT_STUFF string\$

Stuff the given string into the hotkey buffer.

HOT_THING

err = HOT_THING

(key\$, thingname)

An executable thing is started. Things can be part of files which are loaded residently (e.g. QPAC1/QPAC2).

HOT_TYPE

err = HOT_TYPE (key\$)

Returns the type of a hotkey.

-8	last line recall
-6	stuff keyboard queue with previous stuffer string
-4	stuff keyboard queue with current stuffer string
-2	stuff keyboard queue with defined string
0	pick SuperBASIC and stuff command
2	do code
4/5	execute Thing
6	execute file
8	pick job
10/11	wake or execute Thing
12	wake or execute file
-7	Hotkey not defined

HOT_WAKE

err = HOT_WAKE (key\$, jobname)

The job is wakened. If the job does not exist, the function will try to start a Thing with the given name.

The Thing System

Things can be various type of things, e.g.

- a job waiting to be executed from a Hotkey (an executable thing)
- a printer driver which can be used by all jobs
- a piece of code or data for a certain program.

The Hotkey system is implemented as a Thing and is bound together with the Thing system extension in the `hot_rext` file which is an integral part of the Pointer Environment.

The main purpose of the Thing system is to provide a standardized definition format for various things and to make them accessible through the system variables.

Any Thing is identified by its name and thus can be accessed easily from any program.

The Thing system controls installation, usage and proper termination of Things. Cueshell would not run without the Hotkey Thing.

Cueshell keyboard selection keys

Main window

Move	CTRL F4
System control	F1
Catalogue windows list	F2
Memory information	I
Cueshell commands	F3
Jobs list	F4
Hotkeys list	F5
Clock	O
Sleep	CTRL F1
Delete item	D
Refresh directory contents	CTRL F2
One wild card level up	<
One wild card level down	>
Show files with endings only	X
DATA path	D
PROGram path	P
Printer DESTination	SHIFT F4
Change size	CTRL F3

If picked with use Cueshell Hotkey, return to calling job	ESC
--	-----

Catalogue window

Close box	SHIFT ESC
Refresh catalogue(devices)	R
Move catalogue window	SHIFT CTRL F4
Select all	A
List window	TAB
Sort sub-menu	S
View window	V
Change catalogue window size	SHIFT CTRL F3
Scroll up one line	ALT up
Scroll down one line	ALT down
Scroll up one page	ALT SHIFT up
Scroll down one page	ALT SHIFT down
Scroll top	SHIFT up
Scroll bottom	SHIFT down
Pan left one column	ALT left
Pan right one column	ALT right
Pan left one page	ALT SHIFT left
Pan right one page	ALT SHIFT right
Pan start	SHIFT left
Pan end	SHIFT right

System control

All keys marked by underlining>

Keyboard

Close box

ESC

Keys as marked and:

DO set values now

DO (ENTER)

Auto repeat delay down/up

ALT left/right

Auto repeat frequency down/up

ALT SHIFT left/right

Switch job key code down/up

ALT CTRL SHIFT left/right

Mouse

Close box

ESC

Keys as marked and:

DO set values now

DO (ENTER)

Wake up speed down/up

ALT left/right

Acceleration down/up

ALT SHIFT left/right

Hotkey down/up

ALT CTRL SHIFT left/right

Clock

Close box

ESC

Keys as marked and:

Set

DO (ENTER)

Year down/up

ALT left/right

Month down/up

ALT SHIFT left/right

Day down/up

ALT CTRL SHIFT left/right

Hour down/up

SHIFT left/right

Minute down/up

CTRL left/right

Second down/up

CTRL SHIFT left/right

Running clock

C

Catalogue windows list

Close box

ESC

Keys as marked and:

List one line up/down

ALT up/down

List one page up/down

ALT SHIFT up/down

First item in list

1

Second item in list

2

...

...

Memory information

Close

any key or pointer outside

Cueshell commands	
Close box	ESC
All keys as marked	
Configuration	
Close box	ESC
All keys as marked	
Jobs list	
Close box	ESC
Keys as marked and:	
List one line up/down	ALT up/down
List one page up/down	ALT SHIFT up/down
First job in list	1
Second job in list	2
...	...
Hotkeys list	
Close box	ESC
Keys as marked and:	
List one line up/down	ALT up/down
List one page up/down	ALT SHIFT up/down
Hotkey in list	Key
Clock	
Close	any key or pointer outside
All dialog windows	
Close box (if present)	ESC
All keys as marked and	
OK	DO (ENTER)
View window	
Close box	ESC
Toggle view mode	N
Previous file	ALT left
Next file	ALT right
One line up	ALT up
One page up	ALT SHIFT up
One line down	ALT down
One page down	ALT SHIFT down
Sort menu	
Close box	ESC
All keys as marked	