

KingCON

COLLABORATORS

	<i>TITLE :</i> KingCON		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 5, 2023	

REVISION HISTORY

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

Contents

1	KingCON	1
1.1	KingCON documentation	1
1.2	Preface	2
1.3	Changes from v1.1	2
1.4	KingCON compared to CON	5
1.5	Installing KingCON	6
1.6	Installation script	7
1.7	Mounting KCON: as CON:	7
1.8	Editing	8
1.9	Filename- command- and device-completion	11
1.10	Filename-completion	11
1.11	Command-completion	12
1.12	Device-completion	12
1.13	Filenames with spaces	12
1.14	The current directory	13
1.15	Select window	13
1.16	Dropping icons on the console	14
1.17	Other selection-methods	14
1.18	C-flag example	15
1.19	B-flag example	16
1.20	Review-buffer	16
1.21	Differences between buffered text and original output	17
1.22	Memory management - dynamic and static model	17
1.23	Speed on an MC68000-Amiga	18
1.24	Menus	18
1.25	Console menu	19
1.26	Complete-menu	20
1.27	Review-menu	20
1.28	History-menu	20
1.29	Iconifying the window	20

1.30 Options when opening the console-window	22
1.31 Basic template	22
1.32 Current directory and CLI number in the title	23
1.33 Default settings in environment variable	23
1.34 Multiple defaultsettings	24
1.35 All the options	25
1.36 AUTO-option	27
1.37 CLOSE-option	27
1.38 BACKDROP-option	28
1.39 NOBORDER-option	28
1.40 NODRAG-option	28
1.41 NOSIZE-option	28
1.42 SIMPLE-option	28
1.43 SMART-option	28
1.44 WAIT-option	28
1.45 SCREEN-option	29
1.46 WINDOW-option	29
1.47 INACTIVE-option	29
1.48 ALT-option	29
1.49 MINI-option	29
1.50 NOGADS-option	30
1.51 NOREVIEW-option	30
1.52 NOSTYLES-option	30
1.53 NOFNC-option	30
1.54 NOMENUS-option	30
1.55 PLAIN-option	30
1.56 JUMP-option	31
1.57 FONT-option	31
1.58 MAXBUF-option	31
1.59 ASYNC-option	32
1.60 SHOWDIR-option	32
1.61 AUTOICONIFY-option	32
1.62 NOICONIFY-option	32
1.63 IMAGE-option	33
1.64 FNCMODE-option	33
1.65 FASTUPDATE-option	33
1.66 ICONPOS-option	33
1.67 ICONTITLE-option	33
1.68 MENUFY-option	34

1.69 POPSCREEN-option	34
1.70 PREFS-option	34
1.71 KEEPCLOSED-option	34
1.72 To do...	34
1.73 Credits	35

Chapter 1

KingCON

1.1 KingCON documentation

```

*****
*
*      *
*      --- KingCON 1.2 ---      *
*      *
*      --- User Documentation ---      *
*      *
*      --- Copyright © 1993,1994 David Larsson ---      *
*      *
*****

```

Table of contents:

Preface

How to reach me

KingCON compared to CON

Why you should stop using CON

Installation

What does the script do?

Changes from v1.1

New features and fixes in v1.2

Editing

Mastering the command-line

Filename-completion

No more irritating misspellings

Review-buffer

KingCON remembers for you

Iconifying the console

Get lost!

Menus
Features à la carte

Options
Controlling every aspect of KingCON

To do
Some work is still left...

Credits
Thankyou everybody!

1.2 Preface

The program and files in this distribution are freely distributable, but are also copyright (c) David Larsson. They may be freely distributed as long as no more than a nominal fee is charged to cover time and copying costs.

No commercial usage is permitted without written permission from the author. Everything in this distribution must be kept together, in original unmodified form.

The above is generally known as freeware.

Please contact me if you have any comments or suggestions:

E-Mail: f92dala@dd.chalmers.se

Mail: David Larsson
Gibraltarg. 82:150
S-412 79 GÖTEBORG
Sweden

! KingCON needs Kickstart 2.0 or better to run. !

1.3 Changes from v1.1

New features:

*
Default options
can be saved in an environment variable.

* Added new
selection-methods
that don't involve opening a window
when several files match during filename-completion.
This is configured with the
FNCMODE

-option.

- * The memory for the review-buffer can be static if you specify a negative value after
MAXBUF
.
 - * An extra gadget in the titlebar lets you
iconify
the window
quickly using the mouse. The image comes from MUI by
Stefan Stuntz.
 - * The review-buffer has gone through some cosmetic changes, making
it look better when you scroll through it.
 - * You can now insert the current directory, process number etc
in the titlebar by using
substitution-strings
, like the
AmigaDOS Prompt-command.
 - * A custom icon, to use when the console is iconified, can be
defined with the
IMAGE
-option. An exact position on the
screen can be specified with
ICONPOS
.
 - * The title for the icon can be specified with
ICONTITLE
.
 - *
MENUFY
-option to create a menuitem in the Tools-menu when
the window is iconified.
 - *
KEEPCLOSED
-option to prevent the window from opening if text is
ouput to the console while it is iconified.
 - *
FASTUPDATE
-option for increasing the speed of the review-buffer.
Especially useful for 68000-owners. It works by simply updating
the scrollbar more seldom.
 - *
POPSCREEN
-option to bring the screen, on which the window is
opened, to front.
 - * New editing- and manouveringkeys:
-

Control-Left Arrow - Moves to the previous part of a filename.
Control-Right Arrow - Moves to the next part of a filename.
Control-Up Arrow - Moves up one line in the review.
Control-Down Arrow - Moves down one line in the review.
Control-Backspace - Deletes a part of a filename to the left.
Control-Delete - Deletes a part of a filename to the right.

* New menu-shortcuts:

Amiga-J - Enable/disable jump-scroll.
Amiga-. - Show/hide .info-files.
Amiga-W - Enable/disable review-buffer.

- * When completing a filename and no files match your pattern, the devicelist is searched automatically, (just as if you had pressed Shift-TAB).
- * When you Maximize a window, it doesn't cover the first raster line of the screen, so you easily can drag it with the mouse.

Things that should have been in earlier versions:

-
- * RAW: works now.
 - * The console doesn't hang if you have v36 of diskfont.library. KingCON can even run if some disk-based libraried are missing, but some features will be disabled.
 - * The filename-completion doesn't keep any directory-locks.
 - * KingCON doesn't crash when trying to open its window on SID2.
 - * The window is activated when you drop an icon on it.
 - * WINDOW-option accepts 0x in front of the address like CON:
 - * SCREEN-option recognises * as the top screen like CON: does.
 - * Stored command-lines are printed before they are executed when you have specified asynchronous typeahead.
 - * Failing to complete a filename doesn't produce a NULL-character on the command-line.
 - * Wildcards can be used when completing device-names.
 - * The path in the window-title doesn't mysteriously disappear.
 - * KingCON understands which process to signal better.
 - * SetKeyMap works in a KingCON-window.
 - * The correct font is always used when opening a window on a public screen.
 - * The ARexx-commands Queue, Push and Lines can now be used
-

in a KingCON-window. They all use some of the many undocumented features of CON:. For example, WaitForChar() returns the number of lines in the console's internal readbuffer in IoErr().

* Lots of other things that you probably never noticed.

1.4 KingCON compared to CON

The WB2.0 console-handler - CON: - is really good, but it lacks ← some very handy features that make life easier if you use it a lot. KingCON (or KCON) is a replacement that behaves EXACTLY like CON, but it also adds these important features:

*

Filename completion

. You just type the first letters of a filename, press the TAB-key, and - voilà! - the rest of the name is filled in for you (or you will be presented a list of alternative files that you can choose from). This makes typing MUCH faster and minimizes frustrating misspellings. KingCON has many options that let you control the operation of this function. Maybe you prefer it window-oriented, UNIX-like or something else. DOS-commands and devicenames can be completed in a similar way. Some of you might say that the Shell should provide this function instead, but since all editing is managed by the console-handler anyway, this is only natural. And as a "sideeffect", the filename-completion will be available AT ALL TIMES - not only when you are standing at the shell-prompt.

*

Review-buffer

. A scrollbar in the right windowborder lets you browse through all text that has been output to the window at any time. (Very useful when your compiler spits out ten pages of errorlistings.) The buffer can be turned on and off, and saved to a separate file.

*

Intuition-menus

that give you a quick access to the features mentioned above. You can control the size of the window, clear it etc.

* Default settings can be stored in an environment-variable.

*

Jump-scroll

option, which increases output speed a lot.

* The console window can be iconified

and put on the Workbench

as an AppIcon. The window also has a special icon in the titlebar to quickly iconify it.

- * Asynchronous typeahead
, if you want to type in the next command while watching the output from the previous one.
- * Workbench
icondrop
is supported.
- * You can have the
current directory
displayed in the windowtitle.
- * The mouse can be used for
positioning the cursor
on the input line.
- * The text can be any (non-proportional)
font
you like.
Topaz 44 is a killer.
- * The command-history buffer eliminates identical lines, the window is positioned more intelligently etc.

Please note that KingCON does NOT patch CON in any way, it is written completely from scratch. There is a way, however, to mount KingCON as 'CON:'

In this way, every program that use 'CON:' will actually be using KingCON and all the good it stands for.:-> Of course, this is optional.

KRAW: is a replacement for RAW:, and is very similar to KCON:, except it doesn't have any line-editing. See some DOS-documentation for details about the differences between CON: and RAW:.

1.5 Installing KingCON

Simply run 'Installation' by double-clicking on its icon. The script then copies the necessary files to your HD (or Workbench-disk).

When KCON: is installed and mounted, you can try it (at last!) by typing the following in a shell-window (or try clicking here):

```
NewShell WINDOW KCON:,
```

```
NewShell WINDOW KCON:////Shell/CLOSE/MINI/JUMP
```

```
NewShell WINDOW KCON:220//180/100/Shell.%N/AUTOICONIFY/KEEPCLOSED/MENUFY
```

(If you can't find the window after you tried the last example, look at your Workbench or in the Tools-menu...)

When you have found out that KingCON really is worth keeping, you

probably want to replace CON: and use KCON: instead. To do that, you must

```
mount KCON: as CON:
.
```

1.6 Installation script

The install-script does the following things:

- * Depending on which processor you have, 'KingCON-handler' or 'KingCON-handler.020' is copied to L:. Once copied, it is always called 'KingCON-handler'.
- * 'KingCON-mountlist' is copied to DEVS:
- * 'KCON:' and 'KRAW:' are mounted.
- * Optionally, the necessary Mount-commands are added to s:user-startup. They are:
Mount KCON: from DEVS:KingCON-mountlist
Mount KRAW: from DEVS:KingCON-mountlist

1.7 Mounting KCON: as CON:

When KingCON is installed, the DOS-devicename for it is 'KCON:' (and 'KRAW:' for the unbuffered version). If you want that all your shell-windows you open should be KCON-windows, you just change the WINDOW-ToolType of the Shell-icon to say KCON: instead of CON:. (See your Workbench-documentation.) But in this way, you will only get a KCON-window if you start a new shell by double-clicking on the Shell-icon. Furthermore, some programs have their own console-windows and won't let you change them from CON: to KCON:. (Real3D v2.0 is one example).

What you need is something that makes every reference to 'CON:' actually lead to KingCON. This is done by mounting 'KCON:' as 'CON:' instead. BUT, this is normally not possible, since 'CON:' (and 'RAW:') are mounted before the startup-sequence is executed.

This is the solution:

- * Add these lines BEFORE the Mount-commands in the user-startup:
Assign CON: DISMOUNT
Assign RAW: DISMOUNT
- * Change the Mount-commands to:
Mount CON: from DEVS:KingCON-mountlist
Mount RAW: from DEVS:KingCON-mountlist

Now when you boot, every console-window should be a KingCON-window. There is no need to have a special KCON: or KRAW:-device. Enjoy!

!! WARNING !!

Before you try the above, PLEASE make sure that the normal installation was successful and that you can mount KCON: and KRAW: when rebooting.

Make sure that you follow the steps carefully. If something for some reason goes wrong after the Assign-DISMOUNT-commands, you won't have any console at all. A very frustrating situation if you must start your favourite text-editor, in order to correct the mistake, from the Shell... Remember that you have the Workbench-menu "Execute command" as a final rescue!

1.8 Editing

Note: Everything said here is only true for the "cooked" KCON- ← console.

KRAW is "raw" and has no editing capabilities. Also, if you have specified the

ASYNC

-option, things will work a bit different from what you may be used to.

'KCON:' is probably most often used as output for the Shell, but editing works the same if it is a program's private AREXX-console, for example. Even the filename-completion is identical.

I assume some experience of the editing used in 'CON'. Most things are identical, the exceptions from this are marked with a (!).

These keys are used for the line-editing:

Key	Effect

Any "printable" char.	Inserts the character pressed under the cursor, shifting the characters right of the cursor one step to the right.
Return or Control-M	Finishes the line and adds it to the history-buffer, UNLESS it is identical to the previous line.
Alt-Return (!)	The line is finished and added to the history-buffer, but it is not sent to the command shell. This means that it won't be executed.
Control-\	Inserts an EOF-character and finishes the line. This often means that the window is closed, but it's up to the program that opened 'KCON:' to decide.
Control-J or Control-Return	Adds a linefeed.
Right Arrow	Moves the cursor to the right.
Left Arrow	Moves the cursor to the left.
Up Arrow	Displays the previous history-line.
Down Arrow	Displays the next history-line.

Shift-Right Arrow or Cursor to the end of the line.
Control-Z

Shift-Left Arrow or Cursor to the beginning of the line.
Control-A

Control-Right Arrow (!) Cursor to the next part of a filename.

Control-Left Arrow (!) Cursor to the previous part of a filename.

Alt-Right Arrow (!) Cursor to the beginning of the next word.

Alt-Left Arrow (!) Cursor to the beginning of the previous word.

Left Mousebutton (!) Moves the cursor to to the position on the input-line that you clicked on. If you click somewhere else in the window, nothing will happen. Note that the cursor is moved when the button is released and that you must not drag the mouse pointer before you release the button. You can still select text to be copied to the clipboard as usual.

Backspace Deletes the character to the left of the cursor and shifts the rest of the line to the left.

Delete Deletes the character under the cursor and shifts the rest of the line to the left.

Shift-Backspace (!) or Deletes the line to the left of the cursor.
Control-U

Shift-Delete (!) or Deletes the line to the right of the cursor and
Control-K puts the characters in a cut-buffer (NOT on the
clipboard.)

Alt-Backspace (!) or Deletes the word that the cursor is standing on
Control-W to the left.

Alt-Delete (!) Deletes the word that the cursor is standing on
to the right.

Amiga-Delete (!) Deletes the word that the cursor is standing on.

Control-Backspace (!) Deletes a part of a filename to the left.

Control-Delete (!) Deletes a part of a filename to the right.

Control-B or Deletes the whole line.
Control-X

Control-L (!) Deletes the whole line and clears the console.

Shift-Up Arrow or Finds the next history-line whose leading characters
Control-R match the first characters in the line up to the

cursor.

Shift-Down Arrow Clears the line and puts you at the end of the history buffer. (That means that if you then press 'Up Arrow', the last line of the history will be displayed.)

Alt-Up Arrow (!) Scrolls backwards one page in the review-buffer

.

Alt-Down Arrow (!) Scrolls forwards one page in the review-buffer

.

Shift-Alt-Up Arrow (!) Scrolls to the beginning of the review-buffer

.

Shift-Alt-Down Arrow (!) Scrolls to the end of the review-buffer

.

Amiga-V Inserts text from the clipboard.

Control-Y Inserts the characters deleted with Control-K or Shift-Delete.

Control-S Halts the output (if you are listing a directory for example). This could be done by typing any key and then type Backspace to resume output. During filename-completion, the select-window is opened to make it easier to browse through the matching files. See Other selection-methods

.

Control-Q Resumes output again.

TAB (!) Filename-completion

.

Shift-TAB (!) Device-completion

.

Alt-TAB (!) Command-completion

.

Control-D (!) Outputs the files in the directory before the cursor, unless the command-line is empty. In

that case it acts like a breaksignal. See

Other selection-methods
for details.

1.9 Filename- command- and device-completion

The idea with xx-completion (or TAB-expansion) is to make typing faster and more accurate. Once used to it, it will be a great help you don't want to live without.

Filename-completion
Files and directories

Command-completion
AmigaDOS commands

Device-completion
Devices, volumes and assigns

Select window
When several files match

Other selection-methods
Don't like the window?

Filenames with spaces
The use of quotation-marks

Dropping WB-icons
The console as an AppWindow

The current directory
How does it work?

1.10 Filename-completion

If you want to enter a specific filename (or a complete path), with entering the first characters in the filename, then press TAB (or Amiga-F, see

Completion
-menu). If only one file or directory relative to the current directory matches those leading characters, the rest of the name will be filled in for you.

Normally you just enter the first letters of a filename and then press

TAB. If many files begin with the same letters, you can also use standard AmigaDOS wildcards to filter out the file you want. For example, if you want a file that ENDS with the characters '.gif', you type '#?.gif' and press TAB.

If more than one file matches the characters you typed, by default, a window with a list of possible filenames will be opened. (You can change this behaviour, by specifying another selection-method .)

If you didn't type any characters at all before you pressed TAB (or Amiga-F), an asl-requester will be opened for you, where you select your file or directory.

By default, .info-files are not shown. This can be toggled by selecting the menu

```
Complete » Show .info
.
```

1.11 Command-completion

Command-completion works in the same way as filename-completion . The difference is that not only the current directory is searched, but also all the command-paths. Only plain executable files are considered in the search. If your system has many command-paths, this operation may take a while. Command-completion is activated by pressing Alt-TAB or Amiga-M.

1.12 Device-completion

Device-completion is quite simple, and is invoked by pressing Shift-TAB or Amiga-D. All volumes (like 'Workbench:'), assigns ('L:', 'FONTS:' etc) and DOS-devices ('PAR:', 'CON:' etc) are searched for a match.

1.13 Filenames with spaces

If you have completed a filename that includes spaces, quotation-marks will be inserted automatically in the beginning and the end of the word (where necessary). Let's say that you want to enter the following path:

```
My Directory/The Manual.doc
```

You start by entering 'my' and then press TAB. The line will look like this:

```
"My Directory/
```

Now you enter 'the' and press TAB again. The path is now complete:

```
"My Directory/The Manual.doc"
```

Note how one quotation-mark was inserted in front of My Directory and one after The Manual.doc. The path will now be interpreted as one word by the command shell.

If you want to match on a string that contains spaces, you must put a quotation-mark in front of the first word before you press TAB:

```
"my dir<TAB>
```

To sum up, KingCON does the following to determine which filename to complete: If the line contains an ODD number of "'s left of the the cursor, the name is taken from the last quotation-mark to the position of the cursor. Otherwise, the beginning of the word is taken from the character next to the first space, comma, or one of ' = < >. (Dot not included.)

1.14 The current directory

What is this talk about 'current directories' and 'command- ↔ paths'?

'KCON:' is just a DOS-device and has no 'current dir' like the Shell. Is 'KCON:' some kind of shell or what??

The answer is that KingCON keeps track of which process that opened a KCON-window. Every time you invoke filename- or command-completion, KingCON looks at which directory is the current directory for the process that opened the window. Command-paths are also found this way.

If the process that opened the window dies before the window is closed (this could be the case if it is a

```
WAIT-window
.), KingCON keeps
```

the directory that was used last time some kind of completion was activated. Command-paths are lost, though.

I've said it before, but I'll do it again, for the sake of clarity: KingCON is NOT a Shell, despite of the filename-completion and some other functions. No interpretation of what you type is done. To the programs that use 'KCON:' for input and output, it behaves just like 'CON:'.

1.15 Select window

When any of the completion-functions has produced more than one ↔ match,

a window with a list of alternatives will be opened. (Note that there are

```
other selection-methods
, if you are more used to how some UNIX-system do
```

this.) You select the file/command/device you want by:

- * Clicking on it and then choosing 'OK' or
- * Double-clicking on it or
- * Browse through the names with TAB and Shift-TAB or Up- and Down-Arrows, then press Return.

To cancel, you:

- * Press 'Cancel' or
- * Close the window or
- * Press ESC.

1.16 Dropping icons on the console

Another quick way to insert a filename on command-line is to drag one or more icons from the Workbench over the console window and drop them there. The complete path for the icon, including quotation-marks (when necessary), will be printed.

Holding down the Control-key while releasing the mouse-button inserts the pathpart of the filename. (If you have the icon "Work:Stuff/File", only "Work:Stuff/" will be inserted.) Holding down the Alt-key will insert the filename. ("Stuff" in the previous example.)

This is only possible if the window was opened on the Workbench-screen.

1.17 Other selection-methods

When more than one file matches what you have typed, you must choose between the different alternatives in some way. By default, KingCON opens a window with a list of alternative files. Even if this can be fully operated with the keyboard (you don't have to reach for the mouse), you might prefer another way to present the list.

The

FNCMODE

-option lets you set a combination of "flags" that control what KingCON does when multiple files match you description. Each flag consists of one letter, and is typed directly after FNCMODE.

The flags and their function are:

- W - Opens a window with a list of filenames. This is the default.
- C - The leading characters that are common for all the alternatives are inserted when you press TAB for the first time.
What happens if you press TAB a second time (before pressing any other key!) is determined by the other flags.
(See example

.)

- B - After pressing TAB the first time, subsequent TABs or Shift-TABs browse you through the alternatives, forwards and backwards respectively. Pressing any other key than these exits the "browse-mode". Note: This flag cannot be combined with the W-flag for obvious reasons! (See `example` .)

Browsing through many matching files just using TAB can be quite tedious. You can at any time press Control-S to open the

`select-window`, and you will get more overview.

- L - Always outputs a list with all the alternatives on the console.
- S - Silent mode. The display doesn't flash when a completion fails.

The real strength with the

`FNCMODE`

`-option` comes when you combine the

flags to suite your needs. I suggest that you experiment to see what you like best. Some useful combinations that you should try are:

- WC - After the first press on TAB, the common characters are inserted. Pressing TAB again opens the selection-window. Try me! (Assumes that you have mounted KCON:)
- BC - This mode is very common on UNIX-system. First the common characters are inserted, then you can browse through the alternatives with TAB and Shift-TAB. Try me! (Assumes that you have mounted KCON:)

Note that some combinations are quite useless and very confusing. Only specifying the L-flag will just make KingCON print the alternatives, and that's it. No characters will be filled in for you.

1.18 C-flag example

Let's say that you have these files in the current directory:

```
History.doc
KingCON.doc
KingCON.guide
KingCON-handler
```

On the command-line you type:

```
Delete ki<TAB>
```

The string 'ki' will be expanded like this:

```
Delete KingCON
```

The reason is that all the files that matched 'ki' also had 'KingCON' in common. Note that if you had started with typing 'KingCON' and then pressed TAB, nothing would happen, UNLESS you have specified either the W(indow) or B(rowse) flag.

1.19 B-flag example

Suppose that you have the following files in your current directory:

```
History.doc
KingCON.doc
KingCON.guide
KingCON-handler
```

You begin with typing (on the command-line):

```
MultiView ki<TAB>
```

The string is expanded as follows:

```
MultiView KingCON.doc
```

You can now continue to press TAB to get the next match (or Shift-TAB) for the previous) like this:

```
MultiView KingCON.guide<TAB>
MultiView KingCON-handler<Shift-TAB>
MultiView KingCON.guide
```

1.20 Review-buffer

One of the major differences between a CON-window and a KCON- window is \leftrightarrow that the latter has a scrollgadget in the right windowborder, that lets you browse through all the text that has been output to it since it was opened.

When you have done a long directory-listing and want to go back to take a look at the first files that were listed, you just grab the knob and move it up until you find the place you looked for. The arrows can be used to scroll (approx.) one row up and down.

As soon as you type something on the keyboard or a program wants to output something to the window, the "review-mode" is exited and the scrollknob moves to the bottom. This means that if you try to scroll while a file is being Type'd, the text will jump up and down between the position you want to look at and the bottom of the review-buffer. Confusing, maybe, but fully normal.

Differences to original output
Yes, there may be...

Memory management
Dynamic and static

Speed on an MC68000-Amiga
How to make KingCON faster

1.21 Differences between buffered text and original output

The text shown in the review-buffer doesn't always look like ←
the text
that originally was output to the window. Control-sequences to move the cursor, delete characters etc are not interpreted, which means that some programs that do fancy things to their output won't look that fancy when you begin dragging the scrollknob. This is not a bug, just a sacrifice for speed. :->

The output of LhA will, unfortunately, not look very good in the review-buffer. I suggest that you use LhA's option for simple I/O, -Qa. You won't have any progress indication this way, but at least you will see what files that were unpacked.

If you specify the
NOSTYLES
-option, all textstyle control-sequences will
be filtered. This saves some memory and perhaps speeds up output on plain 68000's. The text in the buffer will always be black and white, though.

1.22 Memory management - dynamic and static model

By default, the memory for the review-buffer is allocated ←
dynamically
when something is output to the window. Normally a maximum of 64Kb can be allocated for the buffer before the first lines start disappearing.
The

MAXBUF
-option lets you specify another limit, if you should find 64Kb too much or too little.

Don't forget that no memory is allocated for the buffer when the console window is opened, because of its dynamic nature. The drawback with this is that you lose memory for every line you write. This can be frustrating if you are debugging a program to see if it frees all memory when it returns. ('Avail', 'Program', 'Avail', you know.) In those situations, you should disable the review-buffer.

A better solution is to make the review-buffer static. Entering a negative value after MAXBUF tells KingCON that all memory for the buffer should be allocated immediately when the window is opened. Note that if all the memory can't be allocated at once, the memory will be allocated dynamically instead.

1.23 Speed on an MC68000-Amiga

You may notice that output is slowed down a bit when the buffer \leftrightarrow is enabled, especially if you have a plain 68000-Amiga. On faster machines, it takes more time just to scroll the window than to manage the buffer, so it shouldn't be a problem there.

If you DO find it too slow, there are five things you can do:

- * Enable the jump-scroll. This can be done with the JUMP -option or by selecting the Console » Jump scroll -menu. Output will be fast but maybe a bit jumpy.
- * Specify the FASTUPDATE -option. The scrollgadget will be updated more seldom, and you will gain much speed this way. I suggest that you try this option before you start disabling the scrollbar completely.
- * Specify the NOGADS -option. Because the updating of the scrollgadget has much overhead, removing it in this way is probably enough to solve the speedproblem.
- * Select the Review » Enabled -menu to disable the buffer temporarily.
- * Specify the NOREVIEW -option.
- * Specify the NOSTYLES -option.

1.24 Menus

The menubar contains the following menus and menuitems:

Console
Closing, resizing etc.

Complete
Instead of using TAB

Review
Enable, clear and save the buffer

History
Enable and clear stored command-lines

Some menus change when the console is in RAW mode. The
NOMENUS
-option
opens a window without a menubar.

1.25 Console menu

Reset (Z) - Clears the console-window and resets all
styles etc. Use this if a program leaves
the window in a complete mess.

Jump scroll (J) - If checked, output speed will be increased
by scrolling the text several lines
at the time. Use the
JUMP
-option
to have this function enabled when the
window is opened.

Iconify (I) - The console window is
iconified
and put
on the Workbench as an AppIcon. If
you specify the
NOICONIFY
-option, the menu
will read "Minimize", and the window
window will just be as small as possible
instead of being turned into an icon.

Normalize (N) - Positions the window at the same place
with the same size as when it was opened.
Can be disabled.

Maximize (A) - Makes the window as big as possible
within the visible area of the screen.
Can be disabled.

Halt (H) - Halts output.

Resume (R) - Resumes output.

About... (?) - Opens a requester telling you about the
current version of KingCON, the author etc.

Close (Q) - The same as clicking in the window's
closegadget or typing Control-\. The
process that owns the window decides if
it should be closed.

1.26 Complete-menu

Filename (F) - Completes a filename or directoryname.
The same as pressing TAB.

Command (M) - Completes a DOS-command. Same as pressing
Alt-TAB.

Device (D) - Completes a devicename. Same as pressing
Shift-TAB.

Show .info (.) - If checked, .info-files will be considered
when KingCON is looking for matches.

See also

Completion

.

1.27 Review-menu

Enabled - If checked, all text output to the window
will be saved in the review-buffer. If
not, you will still be able to scroll
through old text, but no new text will
be added.

Clear buffer - Clears the review-buffer.

Save plain text as... - Lets you select a file to which the buffer
will be saved as plain text.

Save with styles as... - Lets you select a file to which the buffer
will be saved, including control-sequences
for text style and color.

See also

Review-buffer

.

1.28 History-menu

Enabled - If checked, lines typed by you will be
saved in the command-history buffer.
If not, no lines are saved, but you will
still be able to browse through the lines
that are already in the buffer.

Clear buffer - Clears the history-buffer.

1.29 Iconifying the window

A very nice way to quickly get rid of a console window \leftrightarrow temporarily is to iconify it. This can be done by selecting the menu Console»Iconify or by clicking the gadget next to the zoom-gadget in the titlebar. Normally an icon with the same name as the window will be created and put on the Workbench. (You can change the icon's default image and position with the

```
IMAGE
and
ICONPOS
-options. See section 8.)
```

The window can then be brought back by simply double-clicking the icon. If you think that icons clutter up your Workbench, you can specify the

```
MENUFY
-option together with
NOICONIFY
in order to have an extra menuitem
```

in the Tools-menu created for you instead.

The icon's or menu's title is normally the same as the window's. You can change it to something else with the

```
ICONTITLE
-option. Here you can
```

use the

```
substitution-strings
to insert the name of the current dir etc.
```

If you have specified the

```
SHOWDIR
-option, the current directory will NOT
```

be added to the icon. Use the %S or %D substitution instead.

If a window is iconified, it will be opened again as soon as a program outputs text to the console or tries to read from it. The

```
KEEPCLOSED
-option
```

changes this behaviour, so the window will remain closed in most cases. If you have the review-buffer enabled, the output will be saved to it, so you can for example start a long directory-listing, iconify the window, and return later to see what was output.

NOTE!

A very important aspect of this feature is that it doesn't always work. When a program has used a special AmigaDOS-call to find out the address of the console-window it is running in, it is no longer safe to close that window. When you have run such a program (SetFont is one of them), and then try to iconify the window, you will be presented a requester telling you that the window can't be closed in a safe way. You can then select that you still want to iconify it, but you should be very careful when doing this. NEVER iconify the window while running 'More', because this WILL crash your system. Iconifying the window when More is finished works fine, though.

This behaviour is a bit ugly, but there isn't much I can do about it. The standard CON: also suffers from this problem, but maybe you haven't noticed it: The

AUTO

-feature, which is CON's way of iconifying a window (sort of), stops working after running SetFont or More.

So, please programmers, stop using the DiskInfo()-call in order to get the address of the window if you just want to know the dimensions of the console. Instead, use the control sequence called 'WINDOW STATUS REQUEST' (9B 30 20 71). The dimensions will then be returned to you with a 'WINDOW BOUNDS REPORT'-sequence (9B 31 3B 31 3B <height> 3B <width> 72). Of course, the console must be in RAW mode when doing this.

1.30 Options when opening the console-window

KingCON's behaviour can be controlled both by the "filename" ← and default-settings saved in an environment-variable. Read these sections carefully, so you can customize KingCON to suite all your needs. (Well, some of them.)

Basic template
 Specifying position and title

Current dir and CLI
 Displaying them in the title

Default settings
 VERY useful! Save them in an env-variable

Multiple defaults
 Yet another way to reduce typing

All the options
 Know them by heart!

1.31 Basic template

A KCON-window is opened in the same way as a CON-window. The template for the "filename" is:

```
KCON:LeftEdge/TopEdge/Width/Height/Title/Options/...
```

All keywords are optional. If you want the default-value, you write nothing. For example, if you want to specify a title, but no position or size, you write: (Don't forget the quotation-marks if you have spaces in the filename!)

```
"KCON:////My Title"
```

The meaning of the keywords are quite obvious. The default is to open a window that covers the whole horizontal and half the vertical visible area of the screen. The window is positioned below the menubar of the screen if possible. The default title is 'KingCON'.

This differs a bit from CON, which always opens a window in the top left corner of the screen with a fixed height. I think KingCON's way of opening is better. :-)

1.32 Current directory and CLI number in the title

When working with the Shell, you probably want to see which the current directory is without having to issue a special command for this. An easy solution is to use the SHOWDIR -option when you open the console-window. This will insert the current dir after your title.

Another, perhaps more flexible, way of doing this is to put substitution-strings in the title, much like the AmigaDOS Prompt-command. These substitutions also work with the

```
ICONTITLE
-option:
```

```
%S Inserts the complete path for the current directory.
Putting this at the end of the title is the same as
the SHOWDIR-option
```

```
%D Only the last part of the current dir is inserted. This
is very useful together with ICONTITLE, since it doesn't
produce as long strings as %S
```

```
%N The CLI-number of the owning process, if it is a CLI.
Otherwise, a minus-sign is inserted.
```

```
%P The name of the owning process is inserted. Note that
in a CLI, this isn't the same as the name of the
command that is executed, since all commands are executed
by the same process.
```

This example creates a rather instructive title:

```
NewShell WINDOW "KCON:////Shell #N in '%S'/CLOSE"
```

The title can then look something like:

```
Shell #4 in 'Work:Pictures/XXX-Rated/Flowers'
```

1.33 Default settings in environment variable

Apart from old CON:, KingCON offers you many options. Probably you find a certain combination useful, but you don't want to type it every time you open a console. In addition to this, some programs have the CON:-file hard-coded, making it impossible to customize those consoles.

The solution to this problem is an environment variable (see the AmigaDOS-documentation) that contains the options you want to be default, unless you specify something else. There is one variable for KCON: (and CON:) and another for KRAW: (and RAW:). They are called KCON.prefs and KRAW.prefs respectively, and have the same form as the "filename" for the KCON: and KRAW: window. Confused? These examples might help:

Suppose that you want a closegadget on every window you open. Then you should put this command in your user-startup (or issue it before you open a console):

```
SetEnv KCON.prefs /////CLOSE
```

If you now type 'NewShell WINDOW KCON:', it will have the same effect as if you had typed 'NewShell WINDOW KCON://///CLOSE'. That saved some typing, didn't it? Now, if you don't want a closegadget for some reason, you will have to type 'NewShell WINDOW KCON://///NOCLOSE'. You could, of course, delete the variable with 'UnSetEnv KCON.prefs', and you will be back where you started.

The advantage with env.-variables becomes even more clear when you want to personalise your consoles more drastically. The following sequence changes the default size, alternative (zip) size, windowtitle and makes the review-buffer static:

```
SetEnv KCON.prefs 0/0/640/512/ElvisCON/MAXBUF-10/ALT590/462/50/50
```

NOTE 1: Changing the default options may make KingCON incompatible with CON:. For example, a program that opens a window without the CLOSE-option, expects that the window will have no closegadget. However, if you have put this option in the KCON.prefs-variable, the window will get a closegadget, and the program that uses the console may not like that, but it is rather unlikely that it actually will. But you should be aware of it, anyway.

NOTE 2: The options in the env-variables have a lower priority than the options that are specified when a console is opened. In the previous example, the windowtitle will only read 'ElvisCON' if you don't specify another title, like 'KingsTON,Jamaica'.

1.34 Multiple defaultsettings

When you open a console, you can specify another env-variable ←
to be
used as default instead of KCON.prefs. This makes it possible to have a set of different settings for different uses, that are easily accessed with just one keyword.

There are two ways to specify the env-variable. The first one is the

```
PREFS
```

```
-option, that is described further down. The second method ←  
involves
```

much less typing: You simply put the name of the variable after the devicename (CON:, RAW: etc), before the first coordinate for the window. The general form for the "filename" then looks like:

KCON:Variable/LeftEdge/TopEdge/Width/Height/Title/Options/...

Note that the variable-name mustn't begin with a number, or it will be recognised as the LeftEdge-value. Now an example:

Let's say you have created a variable like this:

```
SetEnv BigCON 0/0/640/512/Big/MAXBUF-500
```

If you want to open a shell with this very expensive configuration (it will eat ½ Mb for you!), you type:

```
NewShell WINDOW KCON://///PREFSBigCON
```

or

```
NewShell WINDOW KCON:BigCON
```

or, if you want to add some other options:

```
NewShell WINDOW KCON:BigCON/////BIG!!/CLOSE/JUMP
```

1.35 All the options

Most options have an opposite form within parenthesis. (Marked ← with '!' on this page.) They actually reflect the default-state, and specifying one of them is only useful if the positive form is present in either one of the variables KCON.prefs and KRAW.prefs, or a variable you have specified.

For example, NOCLOSE cancels CLOSE, if that was specified in, let's say, KCON.prefs. Using the negative form in an environment variable is thus useless.

Compatible with CON:

```

      ALT
      Alternative zoom-position
!
      AUTO
      Open console automatically
!
      BACKDROP
      Open as backdrop
!
      CLOSE
      Creates closegadget
!
      INACTIVE
      Not activated when opened
!
      NOBORDER
      No border. That's it.
!
```

```
NODRAG
    Surprise! No dragbar!
!

NOSIZE
    No size-gadget

SIMPLE
    Mark and copy text

SMART
    The opposite

SCREEN
    Open on public screen
!

WAIT
    Wait for user before closing

WINDOW
    Open on already existing window

KingCON only:
!

ASYNC
    Asynchronous typeahead.
!

AUTOICONIFY
    Iconify before opening

FNCMODE
    Controls filename-completion
!

FASTUPDATE
    Update scrollbar in a fast manner

FONT
    Use another font

ICONPOS
    Position for the icon when iconified

ICONTITLE
    The icon's title

IMAGE
    The icon's image
!

JUMP
    Jump-scroll for fast output

MAXBUF
    Maximum review-size, static or dynamic
!

MENUFY
    Create Tools-menuitem
!
```

```
MINI
  Make window small when zooming
!
NOFNC
  Disables TAB-keys for filename-completion
!
NOGADS
  No extra gadgets are created
!
NOICONIFY
  Minimize instead of iconify
!
NOMENUS
  No menubar is created
!
NOREVIEW
  No review-buffer
!
NOSTYLES
  Simple review-buffer
!
PLAIN
  Behave like old CON
!
POPSCREEN
  Bring screen to front
!
PREFS
  Custom prefs-file
!
KEEPCLOSED
  Swallow output when iconified
!
SHOWDIR
  Display current dir in title
```

1.36 AUTO-option

```
AUTO    The window isn't opened until some text is output to it ↔
        or
(NOAUTO) a program wants to read from it. It also gets a close-gadget.
You can also use the
        AUTOICONIFY
        -option for the same purpose
if you want an icon for the window while it is closed.
```

1.37 CLOSE-option

```
CLOSE   The window is equipped with a closegadget.
(NOCLASE)
```

1.38 BACKDROP-option

BACKDROP The window is opened as a backdrop behind all other (NOBACKDROP) windows on the screen.

1.39 NOBORDER-option

NOBORDER The window has no border. Very confusing.
(BORDER)

1.40 NODRAG-option

NODRAG The window has no dragbar.
(DRAG)

1.41 NOSIZE-option

NOSIZE The window has no sizing-gadget or zoom-gadget.
(SIZE) The "Minimize"- and "Maximize"-menuitems are disabled.

1.42 SIMPLE-option

SIMPLE You can mark and copy text from the console. This is the default.

1.43 SMART-option

SMART The opposite of
SIMPLE
. This is the
way the console window was before v2.0 of the OS. If the
window is scaled down and then resized again, you will
lose the text that was printed on the revealed portions
of the window.

1.44 WAIT-option

WAIT The window isn't closed until you type Control-\, select
(NOWAIT) Quit or click the closegadget.

1.45 SCREEN-option

SCREEN <name> The window will be opened on a public screen ↔
 with a
 certain name (case-sensitive!). To open on the screen
 "TERM", you type: .../SCREEN TERM/... or .../SCREENTERM/...
 Calling the screen '*' opens the window on the topmost
 screen. Note that the screen must be a public screen, or
 the window will open on the Workbench.

Also see the

POPSCREEN
 -option.

1.46 WINDOW-option

WINDOW <address> The console is attached to an already open window
 with the address expressed in hexadecimal numbers.
 The menus and gadgets will be disabled for compability.

1.47 INACTIVE-option

INACTIVE The window isn't activated when it is opened.
 (ACTIVE)

1.48 ALT-option

ALT<x>/<y>/<w>/<h> When the zoom-gadget is clicked, the window ↔
 will
 get the position and dimensions described by
 x,y,w & h. For example, ...ALT100/100/400/50...
 will make the window a thin strip in the middle of
 the screen when you click the zoom-gadget.
 The
 MINI
 -option is a faster way to specify
 that you just want the window to be smaller.

1.49 MINI-option

MINI The window is minimized the first time you hit the
 (MAXI) zoom-gadget instead of being resized to fill the
 whole screen.
 More control over the actual zoom-size is obtained
 with the

ALT
 -option.

1.50 NOGADS-option

NOGADS No scrollgadget in the right windowborder is created. On (GADS) slower machines this will increase the speed of the review-buffer noticeably. You can still use the alt-arrow-keys to scroll through buffered text, of course. There will also be no gadget in the titlebar for iconifying the window.

1.51 NOREVIEW-option

NOREVIEW The window gets no review-buffer. No scrollgadget is (REVIEW) created and the Review -menu is disabled. If you just want to gain speed, first try the FASTUPDATE -option.

1.52 NOSTYLES-option

NOSTYLES Control-sequences that control textstyle and -color are (STYLES) not saved in the review-buffer. Buffered text will be plain black and white.

1.53 NOFNC-option

NOFNC (Short for No FileName Completion). The TAB, Shift-TAB ↔ and (FNC) ESC-keys don't activate any completion-function in order to be compatible with CON. Note that you can still use the Completion -menu!!!

1.54 NOMENUS-option

NOMENUS No menubar is created.
(MENUS)

1.55 PLAIN-option

```

        PLAIN The same as
        NOREVIEW
        /
        NOFNC
        /
        NOMENUS
        . The window will
        behave exactly like a CON-window.

```

Example: NewShell WINDOW KCON:///Boring.../PLAIN
 (You must

```

        install
        and mount KCON: first!)

```

1.56 JUMP-option

JUMP Enables jump-scroll. Output speed is increased by scrolling (NOJUMP) several lines of text at once. Note that some programs that do their own jump-scrolling (like 'more') may not work very well with this option. If the window is very small, jump-scrolling is automatically disabled.

1.57 FONT-option

```

        FONT <name.size> Another font than the default-font is used. ↔
        To specify
        a bigger topaz-font you type: .../FONT topaz.44/... or
        .../FONTtopaz.44/... Note that the font must be monospaced!
        I suggest that you use the FONT-option to change the font
        instead of the SetFont-command.

```

Example: NewShell WINDOW KCON:///FontTest/FONTLetterGothic.18
 (You must

```

        install
        and mount KCON: first!)

```

1.58 MAXBUF-option

```

        MAXBUF <size> Sets the biggest size for the review-buffer in ↔
        kilobytes.
        The smallest possible value 4. It is not always wise to
        set the size to a very big value, because your system
        can run out of memory if you output a lot of text to
        the console.
        A negative value makes the buffer static, which means that
        the amount of memory you specified (-size) will be allocated
        when the window is opened.
        See also

```

Memory management

.

1.59 ASYNC-option

ASYNC Enables asynchronous typeahead. Normally, when you press a (SYNC) key while a program ('list' for example) is outputting text to the console, the output is halted until you have finished the your input in some way or another. However, if you specify the ASYNC-option, your keystrokes will be buffered while the output is going on and printed on the command-line when the prompt returns.

If you start stepping backwards with the arrow-keys or do something else that makes it hard for you to keep track of what currently lies in the buffer or where the cursor is, the output is halted and the buffer is printed on the command line where you can continue to edit it.

1.60 SHOWDIR-option

SHOWDIR The current directory is displayed in the titlebar.
(NOSHOWDIR) Two space-characters are placed between the title and the path. You can also use substitution-strings for the same purpose.

1.61 AUTOICONIFY-option

AUTOICONIFY This is exactly the same as the AUTO-option, except that (NOAUTO) an AppIcon is put on the Workbench as long as the window is closed. If you try a lot of PD-programs, it can be convenient to have the following line in the startup-sequence:
Run Enforcer FILE KCON:////Bang!/AUTOICONIFY

This option was called AUTOICON before. (Still works.)

1.62 NOICONIFY-option

NOICONIFY The window will just be minimized instead of iconified (NOICON) when you select Iconify/Minimize from the menu. This is identical to the old NOICON-option (which still works.)

1.63 IMAGE-option

IMAGE <filename> This option lets you change the look of the icon when the window is iconified. <filename> is the name of an .info-file (without .info!) in ENV:Sys/, that will be used to describe the icon.

If this option isn't specified, the file ENV:Sys/def_KingCON.info will be used, if it exists.

An example:

```
KCON://///IMAGEmy_icon
```

The file ENV:Sys/my_icon.info will be used as an icon.

1.64 FNCMODE-option

FNCMODE <flags> Sets the flags that determine how you select ↔ the file you want after you have activated any of the completion-functions. See

Other selection-methods for details.

The following sequence makes KingCON behave more "tcsh-like" (if you don't like that window):

```
.../FNCMODE BC/...
or
.../FNCMODEBC/...
```

1.65 FASTUPDATE-option

FASTUPDATE The scrollbar isn't updated at every line that is output, (NICEUPDATE) which is the default, but in bigger intervals. This enhances the performance considerably on slower machines.

1.66 ICONPOS-option

ICONPOS <x>/<y> Specifies the exact position for the icon in screen coordinates. Note that if another icon is already placed at that position, the icon will be put at a random place on the Workbench.

1.67 ICONTITLE-option

ICONTITLE <title> Specifies the title for the icon (and menu) ↔ when the console is iconified. Use substitution-strings in order to

tell the icons apart easier. An example:

```
.../ICONTITLE Shell.%N (%D)/...
```

1.68 MENUFY-option

`MENUFY` When the window is iconified, a menuitem with the same (`NOMENUFY`) label as the `windotitle` is added to the Tools-menu on the Workbench.

1.69 POPSCREEN-option

`POPSCREEN` Brings the screen, on which the window is opened, to (`NOPOPSCREEN`) the front.

1.70 PREFS-option

`PREFS <variable>` The `env-variable` `<variable>` will be used as `prefs-file` instead if `KCON.prefs` (or `KRAW.prefs` for a RAW console). This keyword is not allowed inside the `env-variable` itself (no recursion). Using `PREFS` is equivalent to putting the variable-name right after the colon in the filename, before the `LeftEdge-value`.

1.71 KEEPCLOSED-option

`KEEPCLOSED` If a program wants to do output or input from the console, (`NOKEEPCLOSED`) while it is iconified, the window will not open. Instead, the text will be saved to the review (if it is enabled) and the read-requests will be queued up. This is useful if you want to open a shell and want it to iconify immediately, without first opening a window. You can then use the following combination:
.../KEEPCLOSED/AUTOICONIFY/...

1.72 To do...

KingCON introduces some very useful features that makes it a worthy replacement for CON. But there are still a few things I want to see in a future version:

- * KingCON needs a preference-editor.
 - * Configurable editing- and function-keys.
 - * A configurable shortcut-menu, that lets you quickly insert commonly used strings. These strings should also be tied to the function-keys.
-

- * The review-buffer could be more intelligent and interpret cursor-moves.
- * A new completion-function that completes words from a textfile that you specify. All the switches for DOS-commands, for example. This should replace the command-completion, which is too slow to be really useful.
- * Localized menus, Kickstart 3.0-specific functions, smaller, faster...

If you have something else that you would like to see implemented in KingCON (or if you find one of those annoying b*gs), feel free to mail me. Who knows, maybe I do something about it! (Be prepared that it might take some time before I can read and answer your letter.)

1.73 Credits

I've lost track of how many people have contacted me and come with great ideas and bugreports. If any of you read this, you should know that I am very grateful for your support.

I can't list you all here, but I must mention some of you:
Anders Hammarquist has done a great job when testing KingCON and sharing his experience on consoles and shells.

Stefan Stunz, the author of the beautiful MUI (Magic User Interface), has contributed to KingCON with the source for the iconify-gadget.

Many thanks!

```

      /-\ David Larsson /-\ Engineering Physics /-\_/_/-----
     /  \_/_-\_/_ /---\_/ \ Chalmers University of Technology / Sweden
    /    \_/_/    \_/_ /-\_/_ /-\_/_ /-\_/_-\_/_ /
   -\_/_/ f92dala@dd.chalmers.se \_/_/ \_/_/ \_/_/ \_/_/

```