
The options for the Windows interpreter are contained in a file named RESOURCE.WIN (or whatever it might be named on the command line). It is basically a copy of the DOS "resource.cfg" with the Windows options added. This is because some, but not all, of the DOS options are also used by the Windows interpreter. A line containing "[Windows]" must precede any of the Windows-specific options. This line may be placed at the beginning of the file, since the Windows interpreter will ignore those options in which it has no interest.

This file must be copied to the user's hard disk so that it can be modified both by the setup program and possibly by the interpreter.

Each option is composed of an option name, followed by "=", followed by an option value which can be an integer, a boolean (true, false), or a text string.

There is only one required option, the title option. A second option, which specifies the synthesizer type, is filled in by the setup program after querying the user. As an example, the file might look like this:

```
-----
[Windows]
.
.
<DOS options>
.
.
title = Eternal Quest XI
confirmmidi=true
confirmdac=true
.
.
<other Windows options>
.
.
-----
```

All options have a default value as indicated below. There are two classes of Windows options: 1) shipping, and 2) debugging. Debugging options are for in-house debugging only and should not be included in the shipping game. Shipping options may be included as needed to customize the Windows interpreter for the specific game. The option names themselves are case-insensitive.

Summary of shipping options and defaults

As noted above, there is only one required option: "title". The others have default values and only need be included if the value is to change. The following is a list of available options and their default values (the -new- and MIDI catch options are not included). Following this is a detailed description of each option.

```
allowResize      = true
allowTrueColor  = true
asyncCheckFreq  = 10
borderPixels    = 5
colorCursor     = false
confirmDac      = false
confirmMidi     = false
confirmRedbook  = false
cursorViewAdd   = 0
DACSize         = 10240
hires           = true
maxCursorSize   = 4
requireDac      = false
reserveK        = 300
scimink         = 1500
```

```
<scimaxK      = 8000
synthType     = basesynth
title        = <blank>
titleBar     = true
wing         = true
wingMinK     = 3000
winPatchDir  = <blank>
wWidth      = 640
```

Shipping options

allowResize

values: true, false default: true

If true (the default), when the user resizes the window, the display will be scaled to the new dimensions. If false, then resizing will simply clip the display.

allowTrueColor

values: true, false default: true

Specifies whether the game can be played with a true color video driver (more than 256 colors, non-palettized). If the user has such a driver and the option is false, then he will not be allowed to continue.

asyncCheckFreq

values: 0 - 60 default: 10

This option controls the frequency with which the Windows event queue will be checked. It is specified in times per second; e.g. the value of 10 means that the queue will be checked 10 times per second. Higher values will increase the responsiveness of keyboard and mouse input at the expense of overall performance. Note that this represents a maximum rate; some activities such as large resource loads can interrupt queue checking.

borderPixels

values: 0 - 20 default: 5

When playing a game under Windows, it is sometimes difficult to cause ego to change rooms since the cursor changes to a Windows cursor when it touches the edge of the window. This option allows you to specify that a black border be displayed around the playing area. The value specifies the width of the border in pixels.

colorCursor

values: true, false default: false

Determines whether color cursors or black and white Windows cursors will be used. Windows cursors are interrupt-driven and move more smoothly than do color cursors whose movement can be affected by resource loads.

confirmDac

values: true, false default: false

If a user has not properly installed a Windows wave audio driver, then sampled sounds will not play, possibly resulting in calls to Technical Support. This option causes the interpreter to check for a DAC driver and if none is found, to display a dialog box informing the user. The user then has the option of continuing or quitting. There is a check box in the dialog box which the user can check to inhibit subsequent displays of

the message.

confirmMidi

values: true, false default: false

Same as "confirmdac", but checks for the presence of a MIDI driver. If there is no MIDI driver, music plays through the PC speaker, and the user is informed of this.

confirmRedbook

values: true, false default: false

Same as "confirmdac", but checks for the presence of the MCI CD Audio driver which allows CD Redbook audio to be played. If there is no CD Audio driver, a dialog box informs the user but allows him to continue.

cursorViewAdd

values: 0 - any default: 0

If Windows black and white cursors are selected, the interpreter attempts to render the SCI colored cursors in black and white. If the result is not acceptable, two- or three-valued cursors may be created in separate views. In order to distinguish these from the corresponding DOS cursor, the Windows interpreter will add the specified value to the view number. For example, if the DOS cursor is 100.v56 and cursorViewAdd=2000, then the interpreter will first look for 2100.v56 and, failing that, look for 100.v56. Note that the maximum size of a Windows cursor is 32 by 32 pixels which, with pixel doubling in 640x480 mode, is the equivalent of a 16 by 16 SCI cursor. The interpreter inspects the dimensions of the requested cursor and if they are both 16 or less, it doubles the dimensions.

DACSize

value: 0 - any (should be multiple of 1024) default: 10240

Specifies the amount of memory (in bytes) which is to be allocated for low-level DAC audio buffering. This value should be as small as possible to increase the responsiveness of changes in audio playback. On the other hand, because the Windows audio driver is not interrupt-driven, this value will have to be greater than it is under DOS to eliminate breaking-up of audio while resources are being loaded. Note that this option is in addition to the DOS "audioSize" option which specifies the size of the high-level audio module's working buffer.

hires

values: true, false default: true

The kernel call KISHiRes under Windows normally returns a true value indicating that it is a hi-res environment. This may be overridden by this option if you wish to disable hi-res.

maxCursorSize

value: 0 - any default: 4

If color cursors are selected, then the interpreter must allocate memory up front for building these cursors. The amount allocated must accommodate the largest cursor which the game will use. This parameter tells the interpreter the size of the largest cursor. If the interpreter encounters a cursor larger than this, it will output a message requesting that maxCursorSize be increased.

requireDac

values: true, false default: false

Same as "confirmdac", but does not allow the user to continue if there is no DAC driver. There is no check box to inhibit message display.

reserveK

values: 0 - any default: 300

Minimum amount of available memory (in K) which will be reserved for Windows outside of the interpreter's memory manager. The value required for games to run properly on limited-memory machines is still under investigation.

sciminK

values: 0 - any default: 1500

Amount of available memory (in K) which will be considered the minimum required for the interpreter's memory manager to run the game. If this amount is not available, then a message box will appear informing the user how much additional memory is required to run. The interpreter will then terminate.

scimaxK

values: 0 - any default: 8000

Maximum amount of available memory (in K) which the interpreter will acquire for the memory manager.

synthType

values: pcspeaker, basesynth (default), highsynth, adlib

Specifies which MIDI device type the interpreter will use. This option is not required, since the setup program will fill it in based upon user input. Note also that values "pcspeaker" and "adlib" are available only for the purpose of debugging. If the user does not have a sound card, then the interpreter will automatically use the PC speaker.

title

values: text up to 30 bytes default: none

This specifies the text which will appear in the title bar of the game window. If this option is not found, you will get the message "Windows title not found", which is a good indication that the interpreter did not find the resource.win file.

titleBar

values: true, false default: true

Specifies whether a title bar will be displayed. Note that if the title bar is not displayed, there is no control menu, so the user must use ALT-TAB or ALT-ESC to task switch.

wing

values: true, false default: true

Specifies whether to use the Microsoft WinG graphics speed-up DLL's. WinG must be installed on the user's system for this to work. This option is only meaningful to the WIN32S versions of the interpreter named SIERRAWS.EXE and SCIWS.EXE.

wingMinK

values: 0 - any default: 3000

Minimum amount of memory (in K) the system must have available external to the interpreter for WinG to run correctly. If this amount is not

available, then the interpreter will not use WinG, even if "wing=true" was set. This option is only meaningful to the WIN32S versions of the interpreter.

winPatchDir

values: text up to 50 bytes default: none

Specifies the directory path to check first for Windows-specific patches.

wWidth

values: 320 - 640 default: 640

Without this option, the interpreter will create as wide a window as is allowed by the user's video driver, up to a max of 640 pixels. If you wish the game to start in a smaller window, then it may be specified with this option. In either case, the user may re-size the window. Note that the height of the window is adjusted automatically to the proper ratio.

New caches

Formerly, the interpreter would allocate and de-allocate memory outside of the memory manager's control via the C++ "new" construct. The result was that it was impossible for the interpreter to know up front how much memory it would require. To control this, a series of memory caches has been set up which the interpreter can allocate up front.

The cache sizes in terms of entry size and number of entries are set up with default values which have been found to be sufficient in the normal case. Should the cache space ever be exhausted, then the required memory is obtained from the memory manager. In this case, if the debug interpreter (SCIW) is being used, then the message "Cache<n> overflow:<y>" will be displayed in the debug scroll box, where <n> specifies the cache number and <y> specifies its default max size. This does not indicate an error, but is just a warning that the indicated cache size should be increased. The message will not be displayed in the shipping interpreter (SIERRAW).

There are four caches for satisfying -new- requests. The first is a "reserve" cache for requests made before the Pmachine starts. These are generally for static objects which last throughout the game, and so they are allocated contiguously from the reserve cache. The size of this cache is controlled by the "reserveSize=" option which currently defaults to 35000. The other three caches are in the form of arrays, each with an array size and an element size. These two parameters are controlled by the "cacheSize<n>=" and "entrySize<n>=" options.

It should never be necessary to change the "entrySize" value, but if a "cache overflow" message is displayed, the "cacheSize" value for that cache may need to be increased. The names of the cache control options and their current default values are as follows:

option	default
-----	-----
reserveSize	= 35000
cacheSize1	= 400
entrySize1	= 32
cacheSize2	= 500
entrySize2	= 512
cacheSize3	= 50
entrySize3	= 1200

MIDI caches

When MIDI data is played, it must be in a page-locked data segment since it is accessed at interrupt time. To avoid allocating and de-allocating page-locked segments during the game, up to ten caches can be pre-allocated. This allows for more than one MIDI file to be present since games often pre-load several "snd" files for a room. Should these not be sufficient, then the debug interpreter (SCIW) will display the message

"midi cache overflow<n>" where <n> is the requested size, and it will then allocate the requested memory. As with the -new- caches above, this does not indicate an error, and the message will not display in the shipping interpreter. If the message appears in testing, then it may be desirable to change the cache size values or to modify the game code to reduce the number of MIDI files open concurrently. The option names and their default values are as follows:

option	default
-----	-----
midiBuffSize0 = 100	
midiBuffSize1 = 100	
midiBuffSize2 = 1000	
midiBuffSize3 = 1000	
midiBuffSize4 = 10000	
midiBuffSize5 = 65000	
midiBuffSize6 = 0	
midiBuffSize7 = 0	
midiBuffSize8 = 0	
midiBuffSize9 = 0	

Debugging options

doPpcTest

values: true, false default: true

When the interpreter starts up, it performs a timing test on the user's video driver with the objective of optimizing graphic performance. This is the "pixels per call" (ppc) test. It may be disabled with this option.

forcePpc

values: 0 - any default: 0

The graphics timing test referred to above ("doppctest" option) results in a value for "pixels per call" (ppc). This option overrides that value if it is greater than 0.

lockAudio

values: true, false default: false

For the WIN32S versions of the interpreter, specifies whether the audio buffers should be page-locked.

lockMain

values: true, false default: false

For the WIN32S versions of the interpreter, specifies whether the memory allocated for the memory manager should be page-locked.

monoDisp

values: 0, 1, or 2 default: 0

Determines how debug messages directed to the monochrome monitor should be displayed. A value of 0 causes these messages to appear in a scrolling window on the screen. Values of 1 or 2 direct them to the debug output port (controlled by the Windows debug application DBWIN.EXE). Value 1 displays the messages as received, while a value of 2 inserts a newline after each.

showMem

values: true, false default: false

Causes information to be displayed concerning available and allocated memory.

showNews

values: true, false default: false

trapClose

values: true, false default: true

Previously, you could always exit the Windows interpreter by hitting ALT-F4, choosing close from the control box, or double-clicking the control box. KQ7 has the requirement that the game always be saved before quitting. Therefore, it was necessary for the interpreter to trap the Windows CLOSE message and set the global g_quit to 2 to allow the game to save before quitting which is now the default behaviour. If this option is set false, the interpreter will not trap close messages and will act as before.

useAudio

values: true, false default: true

If set false, DAC audio output is disabled.

useMidi

values: true, false default: true

If set false, MIDI output is disabled.

wHeight

values: 1 - 480 default: 200

Initial height of game window in pixels. This overrides the automatic width-to-height ratio adjustment at start-up. It also allows the start-up window to be smaller than 320 x 240. Resizing the window will revert to normal behavior.