

**** Run time system changes ****
* *
* In reverse chronological order *
* *

Ver 1.001.097

8/23/93

ChrisSmith

fileCopy was added as a new sub-function to the kernel FileIO function. Two arguments are required: source-file name and destination-file name. A zero return indicates success; a non-zero return indicates failure.

Possible DOS error return codes:

- 2 file or path not found
- 9 file locked
- 13 file specified is a directory or file is read-only or share violation
- 17 file can't be created
- 24 too many files opened
- 28 device is full

Ver 1.001.095

5/25/93

ChrisSmith

The command line argument -? (or any invalid argument) when passed to the non-debug interpreter (SIERRA.EXE) will display the internal version stamp number associated with a game. Additionally, an interpreter-game version mismatch or failure to version-stamp a game will result in immediate (rather than delayed) termination of the interpreter.

Ver 1.001.094

5/25/93

Martin Peters

Changes where made to allow transparent views. Color 254 is now used to represent that the color in the v map should be remaped via the remapBuffer which is set by a kernel call. The remapBuffer is loaded with either a range modification (all color in range x to y are inc/dec by z) or a percent lamination (all colors remaped to best matched of x percent of color.) if no remaping is specified 254 is treated as skip. A depth of transparency was added and works on a priority depth.

Example:

a 254 color view at priority 8 is placed over a picture with both pri of 7 and 3. if the remap has no depth specified then the remap view remaps all of the picture below but, if a depth of 2 is specified then the view will modify the priority 7 or even pri 6 color in the vmap but not colors a 3 or even 5 will be remaped. this allow a depth of field translation.

the call value is kernel 123 and is call with the following parms

call type

- 0 = remap by percent
- 1 = remap by index
- 2 = remap off

if type = 0

follow by
percent illumination
[depth of field]

if type = 1
follow by
start range
end range inclusive
remap quantity (how much to add or subtract from index)
depth of field

if type = 2
no other vars

Ver 1.001.092

4/27/93

ChriSmith

SCI.EXE is now the only version of the interpreter with debug. It can be used to run both file-based games and volume-based games. (Actually, there is a temporary version of SCI.EXE called SCITESTR.EXE which is used to run the tester... it will soon/eventually be deleted when the SCI system classes can once again manage menu bars.)

SIERRA.EXE is now the only version of the interpreter without debug. It can be used to run both file-based games and volume-based games.

The dongle-specific version of the interpreter (BETA.EXE) is defunct. We now use STAMPVER.EXE to encrypt a trackable version stamp into each pre-release version of a game before it is sent out to beta testers, reviewers, etc. SIERRA.EXE will timeout with an error if STAMPVER has not been run against it and the corresponding game volume resource.

Ver 1.001.091

4/13/93

Ed Critchlow

Added support for the ARIA sound board to use EMS memory. The DoLoad function in both fileload.c and volload.c was modified to use EMS memory if sound patch 10 is being used (the ARIA sound patch). Also fixed a bug in memmgr.s function GetHandle which trashed bp. Sound.c was modified in the InitSound function to check if the ARIA driver is being used and if it is call DoLoad. Sci.c was changed slightly to make sure InitSound made a call to ARM before ARM got the rest of memory.

Ver 1.001.090

4/13/93

ChriSmith

ARM audio buffering is always used if ARM is available. The 'audioSize=nX' convention described in version 1.001.086 reverts to 'audioSize=nK' where n is the number of K-bytes of conventional memory to dedicate to audio buffering only if ARM is not available (default is 4K).

An immediate 'done' cue is generated if there is a digital sample for a particular sound number but no midi equivalent and the user is not using a DAC driver.

If a digital sample is specified for the Init of a Sound Object and the user is not using a DAC driver and subsequently a midi resource is specified for the Play of that same Sound Object with no intervening Init... the interpreter will effect a (re)Init of the Sound Object just before executing the Play.

Ver 1.001.89

Mark Wilden

More pre-SCI32 cleansing. Removed all externs, externdefs and protos (except for local functions) from source files and put them in header files. Moved global variables from SCI.C to their appropriate source files, and their declarations from SCI.H/.I to the appropriate header files. Moved declarations from SAVEVARS.H/.I to their appropriate header files, so a source file only needs to include the headers that it's interested in, instead of all the headers that the old SAVEVARS.H/.I depended on.

Split MOUSE.S into MOUSE.C and MOUSEASM.S. Split INTRPT.S into INTRPT.C

Changed most variables that began with uppercase to lowercase. Put typedefs with struct definitions.

Removed all warnings generated by Borland C++ and included a special symbol that BC++ needs. The interpreter is now completely BC++ compatible.

Removed unused global and local variables and functions (saving hunk and a little heap). Put Send() inline instead of calling a function.

Removed unused parameters, such as the palette parameter to DrawPic, the mode parameter to InsertPalette and RSetDanPalette, the realYDim parameter to ScaleDrawCel.

Fixed bug where GetCelWide, GetCelHigh, GetCelXOfs, GetCelYOfs and SetNewDim didn't protect DI.

118 files changed!

Ver 1.001.087

3/29/93

ChriSmith

Audio resources are now built into two distinct Volumes via MAKEMAPS.EXE: RESOURCE.SFX for .AUD and .WAV files (typically sound effects) and RESOURCE.AUD for @*. * (typically speech), #*. * (SOL syncs) and _*. * (BSTAR rave syncs). RESOURCE.SFX should always be copied at install time to the current directory just as RESOURCE.000. RESOURCE.AUD may exist in any directory on any device so long as an 'audio=' line exists in RESOURCE.CFG to locate it.

Note: There is a new MAKEMAPS.EXE (dated 3/29/93 or later) and a new AUDDCOMP.EXE (dated 3/29/93 or later) corresponding to this version of the interpreter.

Ver 1.001.086

3/23/93

ChriSmith

Added support for buffering audio samples to ARM memory. Currently, audio ARM allocation is fixed at 32K with an additional 6K of conventional memory allocated for audio DMA and IO transfers. ARM audio buffering is activated by including the where-file or cfg-file directive 'audioSize=nX' where n is the number of K-bytes of conventional memory to allocate in the event of ARM allocation failure (e.g. audioSize=12X).

Note: There are new audio drivers (dated 3/23/93 or later) corresponding to this version of the interpreter.

Mark Wilden

Added support for using ARM memory separately from resource management. Added functions AltResMemAlloc() to get memory, ARMRead() and ARMWrite() to access it, ARMPFree() to release it, and ARMCritical() as a semaphore to determine if we're in ARM code. ARMRead() and ARMWrite() allow moving memory to/from an offset in the ARM allocation.

Added Get32KEMS() and Free32KEMS() to allow access to 32K of EMS memory that will remain at the top of the page frame and so can be accessed like normal DOS memory.

Ver 1.001.082

3/15/93

Mark Wilden

More pre-SCI32 cleansing. Broken PMACHINE.S up into PMACHINE.C and PMACHASM.S. Broke START.S up into START.C and STARTASM.S. Moved dongle code to DONGLE.S. Did more cleanup as below, including culling and sorting #includes, removing unused variables and functions, and moving externs into the correct header files. Not released.

Ver 1.001.081

3/11/93

ChriSmith

The kernel function 'KDebugStr' will scroll text to a second monitor. It takes a single argument: a near string to display or 1 to select the lower text page or 2 to select the second text page or 0 to clear the current text page.

The 'xlate=' directive can be used to display the English equivalent of foreign text (or visa-versa) to a second monitor.

If a resource can not be found or is incorrectly flagged and the user elects to abort the game, debug (if present) will be activated before quitting.

The precedence of alternate resources relative to patches is dependent upon the location of RESOURCE.MSG in the list of patchDir specifications. Only patches found in 'earlier' or the same directory as the equivalent resource in RESOURCE.MSG will be used.

Ver 1.001.080

3/10/93

Mark Wilden

The main thrust of these changes was to move some functions from assembly files to C files for ease in porting, especially to SCI32. In so doing, I did some cleaning up, fixed a bug or two and added some minor features.

Moved many externdefs out of .S file and into .I files where they belong. Removed LINT_ARGS from many header files, as well as unnecessary 'extern's on function declarations. Changed some function definitions to use ANSI argument lists instead of K&R. Sorted many lists of include files so that duplications can be seen and to avoid order dependencies. Header files should include all the header files they need, and no more. Moved some typedefs from TYPES.H to their appropriate header files. For instance, Obj is now typedefed in OBJECT.H. Passing code through BC++ let me see some local variables that weren't being accessed, which were removed. Added some casts to remove conflicts between signed and unsigned character comparisons, and comparing signed values to -1.

Profiling code was made conditional on PROFILER, which reduces the size of the code when profiling is not used (as now).

If the user misspells the name of an object he is inspecting or breaking on, he can edit the name, instead of using memory location 0.

The send stack now shows the address of a disposed object instead of ???.

Moved some debugging functions from OBJECT.S to DEBUG.C. Moved other functions from OBJECT.S to OBJECT.C. OBJECT.S was renamed OBJASM.S.

GetProperty and SetProperty now abort the program if an invalid property is accessed. This change immediately made Animate() abort because it was still accessing the nonexistent 'palette' property of View. Taking this access out will speed up Animate(). If you need to determine if a property is valid, GetPropAddr will return 0 if the property is invalid. E_INVALID_PROPERTY was added to the error messages. MacGetPropAddr was renamed GetPropAddrMacro so it wouldn't sound like a Macintosh function.

FreeHeap() wasn't saving DI, which sometimes crashed the program after resources were viewed.

Ver 1.001.077

Mark Wilden

Moved PanicMsgOutput from DOS.C to ERRMSG.C. Cleaned out some LINT_ARGS from ERRMSG.C.

ChriSmith

The functionality of RESOURCE.MSG/MESSAGE.MAP has been expanded to permit resources of any type. The MAKEVOLS -m option can be used to build these special files via the MESSAGE.TXT file (which functions identically to the RESOURCE.TXT file). The expanded purpose of RESOURCE.MSG is to serve as a collection of language-dependent resources. Consequently, RESOURCE.000 should contain only language-independent resources. The sequence for locating resources is (1) patch files; (2) RESOURCE.MSG; (3) RESOURCE.000; (4) load error.

Audio and sync resources can be queued up to 10 sequences deep via the new kernel calls (DoSync QueueSync) and (DoAudio Queue). These two calls can be made in any order or interspersed, but there must be one call to (DoSync Start) for the first sync and one call to (DoAudio Play) for the first audio. The (DoAudio Play) call (and optional (DoAudio WPlay) call must follow all related (DoAudio Queue) calls. A call to (DoAudio Stop) will flush all audios queued and (DoSync StopSync) will flush all syncs queued.

(DoSound Stop) called directly or indirectly will only stop a dac sample if the sound number matches the sound number of the sample playing.
(DoAudio Stop) can be called to stop any dac sample playing.

Ver 1.001.076
Mark Wilden

2/26/93

Added rename() function and (FileIO fileRename) kernel call.

Ver 1.001.073
ChrisSmith

1/15/93

.WAV resources (Microsoft WAVE audio files) are now recognized by the interpreter; .CDA resources (CD 'Redbook' audio files) are no longer recognized as such. .WAV resources can co-exist in a game with .AUD and .SND resources. The order of selection for a particular 'sound' number is x.AUD then x.WAV and lastly x.SND.

Base-36 audio and sync files can now be included as patches. The designations 'audio=' and 'sync=' can be used in RESOURCE.CFG to specify all of the paths to scan for audio and sync resources.

Ver 1.001.072
ChrisSmith

1/15/93

Bug fix for the -w command-line option which existed since it was first coded but was only unmasked by changes made in Ver 1.001.071.

Ver 1.001.071
ChrisSmith

1/15/93

New kernel call 'SetQuitStr' (= 135) takes one near string argument (typically a message) which it will display at the DOS screen when the interpreter exits. If multiple calls to SetQuitStr are made, only the last will be honored. The maximum number of characters displayed will be 255.

An unrecognized command-line argument (such as -?) will result in the version number of the interpreter being displayed (and, if it is a 'debug' interpreter, a list of valid command-line arguments). The interpreter will then exit. This feature can be used without any other files being present.

The CD-ROM drive is located via the AUDIO= line in the where or cfg file. If no such line exists, then it is assumed that the current drive is the CD-ROM drive.

Ver 1.001.070
ChrisSmith

1/15/93

Modified PatchDir syntax and all pertinent patch-management routines to permit more than one path specification for locating resource patches. If PatchDir is omitted, just the current directory will be used for locating patches; if PatchDir is specified then the current directory will not be used for locating patches unless it is explicitly named. Directories are scanned in PatchDir-specified order and duplicate resources are bypassed.

CD-ROM audio 'red-book' functionality is re-introduced in part into the interpreter. Currently, the red-book driver must be named 'AUDCDROM.DRV' and must be located in the same directory as the audio driver that was loaded. If no audio driver was successfully initialized then no red-book audio can be played.

Ver 1.001.069
Mark Wilden

1/12/93

Added capability to specify arbitrary characters in message text with \xx syntax.

Trapped | character in edit controls, since it's a control character for text output.

Added (Message MsgGetKey @vals) kernel call to retrieve the module, noun, verb, case and sequence number of the last message retrieved by (Message MsgGet) or (Message MsgNext).

Ver 1.001.068
Larry Scott

12/30/92

Changed end point calculation for container polygons when OPT = 0.

Ver 1.001.067
Larry Scott

12/15/92

Put timing loops in the show styles for picture drawing. They were drawing too fast on 486 machines.

Ver 1.001.066
Mark Wilden

12/14/92

Added new Message kernel functions:

MsgGetRefNoun, MsgGetRefVerb and MsgGetRefCase, which return the appropriate value, or -1 if the message isn't found.

MsgPush and MsgPop which save the current iterator state, so that a sequence of messages can be displayed without interfering with a previous sequence.

Ver 1.001.065
Larry Scott

12/10/92

Fix on avoider for container polygons. Going from one container to another.
See main1.c and test76.bat.

Ver 1.001.064 12/08/92
Larry Scott

Changed sci.c method of getting the resource.cfg directory for Terry McHenry.

Ver 1.001.063 12/07/92
Larry Scott

Changed getpath.c and getpath.h to fix container polygons. Also changed some code for clarity.

Ver 1.001.063 12/07/92
Jack Magne

Terry M. made changes to sci.c and savegame.c where saveDir is set to where the where file is located.

Ver 1.001.062 12/02/92
Larry Scott

Made some changes to getpath.c for container polygons. See Main55.c and test75.bat in the regression test suite.

Ver 1.001.061 11/25/92
Larry Scott

Made some changes to getpath.c for container polygons. In some cases the path generated did not extend as far as was possible. See Main54.c and test74.bat in the regression test suite.

Ver 1.001.061 11/18/92
Larry Scott

Changed the keyboard polling from 6 ticks to 1 tick. This was to prevent animation cycles from going by while there was a keyboard event waiting to be posted. Along with this change is a change to the audio drivers to intercept the keyboard events to signal the audio driver that a keyboard interrupt is in progress. This is because there will always be a small window where the audio driver can be invoked while in the keyboard driver. If this interpreter is used without the new drivers lockups will result!

Ver 1.001.060 11/11/92
Ed Critchlow

Made local coordinates global for dialog box text selection

Ver 1.001.059 11/10/92
Ed Critchlow

Fixed a bug caused by not setting "acc" in kernel.c only

Ver 1.001.058 11/06/92
Jack Magne'

Fixed a bug caused by the previous fix regarding stopUpdated, scaled views.

Ed C.
Created the ability to select words in dialog box with mouse.

Ver 1.001.043
Jack Magne 10/21/92

Fixed a bug where a stopUpdated, scaled view fails to erase itself from the screen when its dispose method is invoked.

Ver 1.001.042
Martin Peters 9/1/92

Added new random number generator to math.s. It now uses a 32 bit seed and returns a 32 bit value. Changed Kernel.c to use the middle 16 bits of the 32 bit random as a multiplier on the range of value. once the range is scaled by the random 16bits it is then shifted left 16 to obtain a random value between 0 and (range - 1) this is added to the low offset to obtain a random value between low and high. Also the seed set and retrieval is done by passing a single 16bit pointer to a 32 bit location @(array size 2) in the first parameter to set or in the third to retrieve the seed.

Ver 1.001.041
Larry Scott 8/27/92

Added default cursor view to the interpreter. If 998.v56 exists, loop 0 cel 0 of that view will be the default cursor. If not the default cursor will be the cursor in the driver.

Ver 1.001.041
Larry Scott 8/27/92

Fixed bug in getpath.c. It concerned a BAP polygon within a TAP polygon.

Ver 1.001.039
Larry Scott 8/12/92

Added error check for returning heap at an odd address.

Ver 1.001.039
Chad Bye 8/12/92

Changed SAVEGAME.C to use PK Implode compression instead of RLE.

Added two new kernel calls: Record and PlayBack. All sci events and mouse movements can now be recorded and saved to a file for playback.

The format for these functions is:

(Record flag <filename>)

(PlayBack flag <filename>)

flag=TRUE turns the function on, flag=FALSE turns it off.

If a filename is not supplied, data is written to sci_evt.mac.

The macro file is a text file which can be edited if necessary.

The format for the file is:

eventPollCnt, event type, event message, event modifiers, event y,
event x

There must be a blank line seperating each event.

These functions will only work in the debug interpreters.

Ver 1.001.038

Mark Wilden 8/11/92

Upped message stack size to five entries.

Ver 1.001.037

Martin Peters 8/11/92

Added Fix to scaling. Changes included from IBMNEXT. changes to scale.c to calculate ne offset values within cells. Changes to CELS.S to use new scaled offset in calculating cel positions and procedures to return current offset values. Finally changed animate to include new calls to these procedures to scale offsets.

Ver 1.001.036

Mark Wilden 8/11/92

Support ME 4 message file format. Support reference fields, where one message points to others.

Ver 2.000.014

Jack Magne

Improved operation of the SetCursor call that restricts the cursor to a rectangular area.

Ver 2.000.013

Martin Peters

Modified animate to scale the x and y offsets of scaled cels so they are drawn relative to a proportional x and y origin point in the cel. This removes the hopping look for cels with movers.

Ver 2.000.011

Chad Bye 7/27/92

Added text codes to support multi-color and multi-font text messages. Two new kernel calls were added for this: TextColors and TextFonts. TextColors and TextFonts are used to initialize the colors and fonts that can be used in text messages. The format for the new functions is:
(TextColors <color 0> <color 1> <color 2> ...)
(TextFonts ...)

The following text codes can now be inserted in any string:
|c#| - Text following this code will be color # in the text colors table.
|c| - Text following this code will be the default window color.
|f#| - Text following this code will be font # in the text fonts table.
|f| - Text following this code will be the default window font.
|| - Prints the character '|'

Changed ERRMSG.C. There is no longer an ErrMsg function. Panic and RAlert have both been changed to accept a message number (which is loaded from INTERP.ERR) instead of a string. These functions also have a variable length argument list, so formatted strings can still be used.

Added many error messages to INTERP.ERR because of the above change.

Ver 1.001.029

ChrisSmith 6/1/92

Audio samples can now follow the same NOUN-VERB convention as Msg resources and/or follow the standard resource-naming convention (e.g. 123.AUD). A makevols-like utility named MAKEMAPS.EXE is used to combine individual audio samples into room-specific .MAP resources which are then built into volumes in the same manner as all other resource types (*.AUD samples are built into 0.MAP). RESOURCE.AUD is generated in the process which can be split across floppies and recombined at install-time just as RESOURCE.000.

CFG/WHERE files now recognize the individual lines audioSize=?K (default is now 8K), audioDisk=YES or NO (default is YES; specify NO if your game does not stream audio from disk -- that will save you half-a-K of hunk) and audioPort=xxx (default is 220 hex; this line only has meaning to Sound Blasters). The parameters 'size=', 'disk=' and 'port=' (which use to be on the audioDrv= line) are no longer recognized.

Ver 1.001.028

Mark Wilden 6/1/92

Fixed ERRMSG.C to allocate larger buffer for preloaded error messages.

Ver 1.001.027

Mark Wilden 5/29/92

Preload error message for ARM checksum failure to avoid memory allocation while displaying it.

INTERP.ERR changed.

Larry Scott 5/28/92

Added new function to PalVary kernel call. PALVARYTARGET will allow a custom palette to be inserted into the target palette.

The DrawControl routine now looks at the NoShowBits flag instead of the PicNotValid flag.

Ver 1.001.026

Mark Wilden 5/27/92

ARM now keeps its pageMaps in hunk instead of heap, saving even more heap. Code was generalized to remove duplication, which saves hunk.

Mark Wilden 5/28/92

Changed name of KCheck() to KResCheck().

Larry Scott 5/26/92

Changed kernel call to KPicNotValid to set NoShowBits instead of PicNotValid. Also KDrawCel now looks at NoShowBits to determine if ShowBits is to be called. The SCI programmer show not be modifying PicNotValid.

Mark Wilden

5/22/92

Ver. 1.001.025

Improved ARM checksum failure message. If ARM detects that the data it retrieves from EMS/XMS/Extended memory is not the same that it wrote, it aborts with a message telling the user to run Install and choose not to use his extra memory.

INTERP.ERR changed.

Mark Wilden

5/20/92

Ver. 1.001.024

Fixed ARM bugs introduced by Jeff in the MASM 6.0 conversion and Tom Shimada in conditional compile for NEC9801. The first reversed an error condition when running with no memory manager. The second didn't store data when using XMS (as with QEMM with the FR=NONE parameter).

Made ARM pagemaps dynamic, which will save at least 2K of heap.

Removed VRAM, which will save another K.

Generalized some ARM code, which will help hunk.

Made ARM checksumming always on. If ARM detects that the data it stored is bad, it will Panic out, instead of letting things proceed with the bad data.

Larry Scott

5/15/92

Ver. 1.001.023

Changes to allow PalVary work for save, restore and restart.

Mark Wilden
5/12/92

Ver. 1.001.022

Fixed several bugs in FILEIO.S where drive letters were assumed to be in lower case and where RGetFreeSpace didn't return default drive space when passed 0.

Jack Magne
5/11/92

Ver. 1.001.021

Added a special effects cursor that can be used for various effects like magnification and see-through.

Larry Scott

5/11/92

Ver. 1.001.021

Added code to make PalVary work under the ega640.drv driver. The change will always happen instantly under EGA. Also added the variable NumberColors that's set by the video driver.

ChrisSmith

5/11/92

Ver. 1.001.021

Added three new where/config file arguments that are placed on the audioDrv= line: size=nK (default n is 4) disk=x (default x is YES which means audio samples can be played directly from disk; NO will cause the audio drivers to purge direct-disk play support, saving about 1K memory space) port=x (default is 220 hex).

Added static buffer to audio support that does not span a 64K boundary so that continuous DMA can be utilized for sample playback. The actual buffer allocated depends on the size= argument as described above.

Added kernel function KCheck (works just like KLoad but returns TRUE/FALSE instead of handle if the specified resource(s) can be located -- no read will take place).

Larry Scott

5/4/92

Ver. 1.001.021

Added debug commands k and K which break on a specific kernel call. k is break once and K is a sticky break. When you enter k or K you will be prompted for the specific kernel call number. i.e. 113 is KPalVary.

Corey Cole

5/5/92

Ver. 1.001.020

Fixed a bug in script.c which caused crashes in Mixed-Up Mother Goose and potentially corrupted memory in other games.

Larry Scott

5/4/92

Ver. 1.001.020

Expanded the functionality of CheckFreeSpace. See below for a description of the kernel call:

John Magne
4/21/92

Ver. 1.001.013

Fixed bugs to allow the Pal Varying scheme to work smoothly on room changes.

John Magne
4/20/92

Ver. 1.001.012

Fixed minor bug in the Brush sub function of routine of DrawPic. Was causing occasional lockups on room changes.

John Magne
4/17/92

Ver. 1.001.011

Fixed a bug in midi.s that was resulting in writing data to memory reserved for the operating system. This was causing various QEMM exceptions. We are still using the pre Masm 6.0 version of midi.s

John Magne
4/15/92

Ver. 1.001.010

Retrieved the last version of midi.s and midi.i that used Masm 5.1 to attempt to remove bugs created by the conversion to Masm 6.0.

4/6/92 - John Magne

Fixed problem that resulted in overlays being drawn backwards.

4/1/92 - Mark Wilden

Added MsgSize function to Message kernel call. The calling syntax is (= size (Message MsgSize module noun verb case sequence))
The function returns the size of the buffer needed to store the message, including the trailing NULL. A blank message returns 1. A missing message returns 0. Stage directions (all caps inside parentheses) are included in the length

Fixed Panic message.

3/15/92 - John Magne

Added support for the dynamic scaling of views. Thus views can be sized depending upon their position on the screen or under direct control of the programmer.

3/3/92 - Jeff Stephenson

"Make"ing the system now does a number of version-control related tasks. First, it creates the file 'info.c' including the VCSID of the person making the interpreter, the date/time of the make, the version number, the path of the directory in which the make is executing, and a comment from the file .\comment\comment. Make is then invoked recursively after this is created and since all interpreters depend on this file, all interpreters are then rebuilt.

Once the interpreters have been successfully built, all version controlled

files in .\vcs are stamped with the current version number. While this takes some time, it allows the source for any module to be retrieved by version number.

Two utilities are required for this process and should be in your execution path (they are present in i:\interp\sci\ibm):

inc_it: Increments the last three digits of a dotted version number in a specified file.

```
inc_it version.dat
```

increments the version number in 'version.dat'.

oneline: Writes its parameters to stdout as a single line. Any string bracketed by '@' is interpreted as the name of a file which is to be included in the output line. Thus, if the file 'test.dat' consisted of the string "is a", the command

```
oneline This @test.dat@ test
```

would produce

```
This is a test
```

on stdout. The '\' character works as an escape character, allowing an '@' to be included, if necessary.

Borland C++ source code for these utilities is in i:\interp\sci\tools\utils.

The assembly language preprocessor for Intel code (as.exe and ap86.exe) are now obsolete. All assembly source code for the interpreter has been converted to use the MASM 6.0 structured assembly constructs, which are far superior.

***** PLEASE use these constructs rather than labels and jumps! *****

If you don't have a copy of MASM 6.0, get one from Larry.

8/08/91 - ChrisSmith

Restructured map file layout: All resource entries are ordered by type and subordered by id; bits pertaining to type are removed from each individual entry allowing ids to range up to 65536; index table at head of map file indicates where each resource type group starts -- its format is 1-byte resource type followed by 2-byte offset from start of map file to first resource of that type -- indexes are also sorted by type but can not be indexed into since gaps may arise from no resources of a specific type; the last index is followed by a byte of 255 and 2-bytes of map file size (so that a specific resource entry can be found by binary searching its type group).

Auto-patch table generation: A table of (byte)types/(word)ids is generated upon game start or restart so that resource patch search begins with a memory-resident table look-up rather than a disk(disc in the worse case of CDROM) search. A .cfg file entry of patchDir=directory indicates where feed for the table generation may lie (no patchDir entry defaults to current directory).

7/11/91 - Mark Wilden

Added MSG resource type. These are used in text versions of games to

store text messages that are mirrored in sync and audio files in CD versions. Added

```
GetMessage(int module, int talker, int msg, char* buffer)
```

to access this resource from the interpreter and a KGetMessage() wrapper to access from SCI.

5/27/91 - Pablo Ghenis

Internal rearrangement of functions and variables in SCI modules (both C and assembler) in order to improve code reusability, size and speed (yes. there IS a free lunch!). This set of changes affects i:\interp\sci, which contains programs SCI and MAKEVOLS. To get the full benefit of these changes, they should also migrate as soon as possible to i:\interp\sci\sc and i:\interp\sci\parscomp.

The following conventions are now observed (anyone linked with the generation and compilation of non-compliant code will be summarily executed!):

1. There are no extern declarations within .c or .s files, only within .h and .i files. There are no duplicates.
2. Each public symbol (variable or function) is declared in a header file with the same name as the source file that provides it. For example, Panic is a procedure provided by start.s, so it is externally declared in start.i (for .s files to include) and also in start.h (for .c files that invoke Panic).

This will make it easier to find things in the interpreter directory, since all one has to do is type:

```
GREP foo *.h *.i
```

to find out where "foo" is defined, since the source file is guaranteed to have the same name as the header file that declared it.

3. All symbols that exist in the standard C header files (ie. those provided by Microsoft C) are declared in a file that has the same name. For example, our SCI directory contains a file named STDLIB.H that contains the prototype for atoi(strptr s), since this is where atoi is found in the standard header file set. To comply with (2) there is now a stdlib.c that resolves all externals declared in stdlib.h

This will make it much easier to share source modules among the different tools we use, most of which include the standard header files (except the interpreter!). For example, if we now include our own version of string.h we are guaranteed NOT to redefine symbols provided by other header files, unlike the previous situation where the order of inclusion was critical to make sure that we "won" in case of redefinitions.

4. All symbols that are local to a module are now declared "static near", which makes code smaller AND faster. All external symbols are declared global. Every symbol is explicitly tagged as EITHER global OR static near, to improve readability. The interpreter is now about 2300 bytes smaller, and a bit faster as well.
5. All programs now compile and link with ZERO warnings in i:\interp\sci. This will prevent new warnings from being overlooked, since there are not supposed to be any.

3/14/91 - Mark Wilden

Updated from foreign language interpreter. Changes involved calling StrSplit where appropriate, different parameters to TextSize, StrSplit and GetFarText, adding NEC9801 conditional code. Files changed: DEBUG.C DIALOG.C FARDATA.C KERNEL.C LANGUAGE.C MENU.C TEXT.C WINDOW.C KERNDISP.S FARDATA.H KERNEL.H LANGUAGE.H TEXT.H. STRSPLIT.C/.H was renamed LANGUAGE.C/.H.

3/1/91 - Mark Wilden

soundDrv entry in WHERE and RESOURCE.CFG enhanced to allow a hex number following name of the driver (with whitespace in between), which initializes soundPort variable.

2/22/91 - Mark Wilden

memoryDrv entry added to WHERE file and RESOURCE.CFG. If it's NO or no, ARM isn't used. This for the benefit of INSTALL. The -a flag is still available. Changed SCI.C and RESNAME.C.

2/11/91 - Mark Wilden

VGA320.DRV now returns 256 on Detect (instead of -1).

1/30/91 - Mark Wilden

Support for multiple extensions in where files added. OpenResFile added to RESNAME.C which looks for resources with various extensions as well as in multiple paths. MAKEVOLS reads a where file to find its resources. Routines common to FILELOAD and VOLLOAD moved into RESNAME. All programs default to new resource names. SCI*.EXE defaults to using ARM, and -a turns it off. Removed useNewResNames parameter from ResNameMake, which accounts for most of the files changed: DOS.C RESOURCE.H RESNAME.C VOLLOAD.C FILELOAD.C SCI.C DEBUG.C for SCI*.EXE. MAKEVOLS.C MAKEVOLS.MAK for SCI.EXE SC.DEC UPDATE.C VOCAB.C SC.H INPUT.C OBJCODE.C CLASS.C TEXT.C SC.C for SC.EXE VC.C DC.C CTC.BAT GC.BAT MKVOCAB.GAM MKVOCAB.SYS for parse tools.

1/16/91 - Mark Wilden

Support for extended memory and XMS memory ARM types has been added. Files changed: ALTRES.C. Files added: XMS.S EXTMEM.S.

1/14/91 - Mark Wilden

A different way of naming resource files was added (0.v56 instead of view.000, e.g.). Wherever possible, programs call ResNameMake() to create the name based on the presence of the -N flag. Files changed: FILELOAD.C DEBUG.C RESOURCE.H SCI.C VOLLOAD.C RESOURCE.C DOS.C RESNAME.C (added) for SCI*.EXE; UPDATE.C VOCAB.C TEXT.C OBJCODE.C CLASS.C INPUT.C SC.C SC.H SC.DEC for SC.EXE; MAKEVOLS.C, VC.C, DC.C, CTC.BAT, GC.BAT, GC.LSP (!), MKVOCAB.GAM and MKVOCAB.SYS.

1/09/91 - Mark Wilden

ARM supports multiple alternate resource memory types. In addition to VRAM, EMS support was added. Files changed: ALTRES.C VRAM256.S RESOURCE.H. Added EMS.S and ALTRES.I.

1/04/91 - Mark Wilden

All references to VRAM have been moved into ALTRES.C, the alternate resource memory manager. This involved changes to START.S, RESOURCE.C/.H and DEBUG.C.

Command line processing is now done -before- main() and indeed before InitMem(). This means that any variables set by command line flags must be initialized at definition (e.g. int i = 0;) to prevent them being placed in the BSS segment, which is cleared by InitMem(). Changes to START.S and SCI.C.

12/06/90 - Mark Wilden

UPDTOOLS.BAT now archives the old tools to OLDTOOLS.LZH. Use LHARC to extract them if necessary.

11/30/90 - Mark Wilden

DrawStatus() now takes two arguments for foreground and background color.

11/29/90 - Mark Wilden

Added InitMenu() call to main() and KRestartGame() so that theMenuBar's rectangle is initialized even if no menu items are ever added. This is because the status bar uses theMenuBar's rectangle.

10/31/90 - Mark Wilden

Fixed bug in text.c (GetLongest) that made certain text messages print with an extra blank line.

10/22/90 - Bob H.

Extended floppy disk support added. At this writing it is disabled unless '-f' option is passed to SCIVXXX. First stage support will SCAN all drives (A-F) for the required volumes. This will allow users with MULTIPLE drives of comparable capacity to use multiple floppies. This will work for SINGLE PACK games (3.5 or 5.25) only. Subsequent support will handle DUAL PACK games.

FILES MODIFIED:

sci.c - Parsing of command line switch. declaration of variable "oadAccessMode".

sci.h - EXTERN statement for oadAccessMode.

fileio.s - "Exists" function. Error trapped for drive scan.

dos.c - Relocated GetGameDisk function (from volload.c), modified to request a particular drive. Added OpenAnyDrive() function.

fileio.h - Added various prototypes and the OAD_??? defines.

volload.c - Re-worked some of DoLoad to use new OpenAnyDrive call and modified GetGameDisk().

10/12/90 - Mark H.

MemoryInfo function selector added called TOTALHUNK. Also system global called hunkAvail.

files changed:

kernel.c

memmgr.s

memmgr.h

kernel.sh

10/11/90 - Mark H.

Some additions from the JONES interpreter...Center Justification in printf and prioritized dialogs.

files changed:

printf.c

dialog.c

grtypes.h

10/01/90 - Mark H.

The kernel call Lock has been added to lock and unlock resources. This is to allow the application programmer to decide which resources should be locked or unlocked. Two new function selectors have been added to the Memory kernel call. They are MReadWord and MWriteWord for referencing and dereferencing addresses.

files changed:
kernel.c
kernel.h
kerndisp.s
kernel.sh

09/17/90 - Mark H.

VOCAB resources are now all unlocked, except for the character translation vocab which is about 2K. Text resources are also unlocked and the PurgeLast function in resource.c has been changed to purge resources on a prioritized schedule instead of just least recently used. This change gave back at least 40k of hunk on HQ2 with the only drawback being a slight pause when parsing a sentence or printing text. RedoStopped in animate.c was also changed to allow changes in update status (from stopped to forced or startUpd) in views that are behind Dialogs. This fix is not the most desirable way to do it, but HQ2 needs it now and substantial changes to animate would be needed to fix it right.

files changed:
animate.c
resource.c
nlparse.c
nlsaid.c
nlvocab.c
nlvocab.h

08/30/90 - Mark H.

Added Sort kernel call that implements a simple bubble sort in C based on a scoring function passed from SCI in a Code Object.

files changed:
selector.h
kernel.c
kerndisp.s
selector.new
kernel.sh

8/25/90 - Bob

Relocated palette matching code from PAL256.c to GRAPH256.s. This move was indicated by the need for greater speed.

8/19/90 - Mark Wilden

Increased size of 'where' file buffer in SCI.EXE to allow for up to 64 char file name.

8/14/90 - Bob

VE256 is now under version control. All changes to this tool MUST be coordinated through Dan Foy OR Corinna. The version

control directory is voll:interp\sci\tools\ve256. In this directory is the proper batch file (vcscfg.bat) to get access to this project. Please see me if you have any questions.

8/4/90 - Bob

The following files are affected by changeable view palettes

CELS16M.S

- Added code to handle remapping. Bulk of code in GetCelPointer. DrawCel NOW expects a pal(ette) argument for 0 to 7. 0 = NO MAPPING 1 - 7 selects appropriate palette
- Monochrome is NOT supported at this time but will be using palette data 0

DIALOG.C

- Added an immediate 0 to the only DrawCel call in this file.

KERNEL.C

- Changed KDrawCel to pass along an optional 7th argument. If this arg is not present it passes 0 to DrawCel.

ANIMATE

- All DrawCel calls are passed the value of s_palette. Also added another element to the AniObj structure (AniObj.pal) for lastCast use.

SELECTOR.H

- Added s_palette as selector 91

DRAWCEL.H

- changed prototype