

Sculptor

RELEASE NOTES

Version 2.0b

15th January 1990

- For Existing Users Only - CHANGES WITH IMPACT ON EXISTING SYSTEMS

1. **spp:** The case statement now falls through to the following case statement if a **break** is not encountered. In version 2.0a, the **break** was erroneously assumed. Existing programs may require insertion of **break** statements to preserve program flow.
2. **spp:** the special **CONTINUE** label is now recognised and is a reserved word. Any occurrence of **CONTINUE** in an existing program will have to be changed if the program is to be compiled with **spp**.
3. **menu:** The settings for the **autohelp** and **autocr** declarations do NOT now carry forward from the initial menu into sub-menus. The default menu settings will apply in sub-menus unless the **autohelp** and **autocr** declarations are used in those menus.
4. **menu:** The **ESCAPE** key now exits the menu and does not select the current option.
5. **menu:** The \$ sign is now retained within the command line and is passed on to the called program.
6. **decprint & decvdu:** The NULL character (\$00) is now correctly processed as \$00 when output as hex using the **-h** option to **decprint** or **decvdu**.
7. **sage:** The interrupt handling under DOS systems has been improved, and it is now possible to interrupt a process at any time it is safe to do so, not just when at an input field.
8. **sage:** When the **lock** command is used on a file, any attempt to insert a record to the file by *another* user will wait until the lock is released. If the process which placed the file lock issues an **insert** or **delete** command, the file lock will be removed before the command is executed. The file lock will **NOT** be automatically re-instated after the command completes. **A write command will remove the file lock if the key values have been changed.**
9. **sage:** When clearing the prompt line, only the text itself is cleared, not the whole line. Programs which write to the prompt line and assume it has been cleared will require some modification.
10. **sage:** The **put #0** command now uses the current video attribute under DOS.
11. **sage:** The **exec** command no longer clears screen line 25 prior to performing the operating system command under DOS.

12. **sage:** When using redirected input, the input stream is now read character by character unbuffered. This will allow **exec** commands within that program to use the redirected input stream correctly.
13. **sage:** The block option line now recognises the SPACE BAR as cursor right movement and does NOT execute the highlighted option.
14. **sagerep:** Statements used within the **!final** declaration now use the last record read from the driving file, not the second last record.

ENHANCEMENTS AND NEW FEATURES

1. **spp**: A number of processing limits have been increased - these are:
 - nested **ifdefs** - increased to 30
 - nested **while** or **for** - increased to 40
 - **case** within **switch** - increased from 40 to 128
2. **describe**: The **undelete** menu option now continues to work after an **edit**.
3. **describe**: A .d file is always saved to the directory from which it was loaded.
4. **pdes**: The long listing (-l) option now allows a full line for each entry.
5. **sp**: The maximum number of **ltemp**s that can be loaded has been increased to 300 from the previous maximum of 200.
6. **sp**: Manifest constants are now fully dealt with by **sp**, so that conditionally compiled code may be handled correctly. The **-D symbol [= value]** syntax may be used on the command line to declare and optionally define the value of a manifest constant.
7. **eq1**: The menu now returns to the query screen after an option has been run.
8. **cf & cr**: The maximum length of an input line has increased to 1024 characters.
9. **vdu parameter files**: An entry has been added to provide support for terminals which use embedded attributes. Currently this is only used by Sculptor when displaying titles in **sp**, **describe**, **menu** and **sage**.
10. **sagerep**: The **put** command will output the system end of line sequence. When outputting to channel 0 an additional carriage-return will be added to the output stream. This will ensure that an end of line is properly placed when the terminal is operating in raw mode.
11. **sage & sagerep**: The **lhandles** declaration has been implemented to provide the ability to open up to 32 Sculptor files and 16 sequential files under MSDOS versions 3.3 and above. The command has the following syntax:

!handles number

Where *number* is the maximum number of DOS file handles to allow in the range 21-255. Each normal keyed file require two handles, each index only keyed file requires one handle and each sequential file channel requires one handle.

The total number of file handles cannot exceed that set by the FILES= parameter in the CONFIG.SYS file. Any attempt to set lhandles to less than 20, greater than 255 or greater than the FILES= setting will be ignored.

This declaration should only be used if your program requires to open more than 8 Sculptor keyed files, or if you are getting an operating system error number 4 (not enough file handles). This declaration is ignored under all other operating systems and under versions of DOS prior to 3.3.

12. **sage:** When writing to a terminal using put, the Ignore protection and disable scroll sequence (vdu parameter file, entry 19) is issued prior to placing the characters and the Honour protection and disable scroll sequence (vdu parameter file, entry 20) is issued after the characters have been placed. This allows put to place characters over protected fields on the VDU.
13. **sage:** In option block mode, the option line wraps from the last option to the first.
14. **sage:** a program which opens a file in read only mode may benefit from improved performance if the file is locked, especially if the operating system uses a daemon to implement record locks.
15. **sage & sagerep:** The remove() and chdir() functions now remove trailing spaces from their arguments.
16. **sql:** The data dictionary field "Hide from SQL" has now been fully implemented according to the following rules:
 - a. In "select *" the hidden fields are not shown and no error is generated.
 - b. A hidden field gives an error number 96 when accessed directly via the where, order by or select clauses.
17. **sage:** Full support has been added for the extended character set between ASCII 128 and 255 for both display and input.
18. **sage:** The redrawing of lines has been optimised to increase the speed of redraw when removing overlaid screens.
19. **sage:** Highlighted fields are now redrawn highlighted when an overlaying screen is removed.
20. **sage:** The hline and vline commands can now deal with one or two character lines.
21. **sage:** The input command now correctly deals with decimal point characters other than . (full stop) as defined by lcf.
22. **sage:** A new environment variable SCCOUNT has been added to control the recognition of special input characters. SCCOUNT may be set to a number in the

range 1 (the default) to 20. This controls the length of time an input routine will wait to see if a special character is being input. On most systems this environment variable should not be used. Some terminals, however, send their special key sequences very slowly. Increasing the **SCCOUNT** value will allow special character strings to be recognised as a single key press.

IMPORTANT NOTE: On some systems, timed input cannot be used and the **SCCOUNT** environment variable should be set to zero. This will prevent the use of the **ESC** key as the end of input key. The "Sent by End of Input and MODE key" entry in the VDU parameter file **MUST** therefore be set to something other than **ESC**. When using the development system programs, **sp** and **describe**, the screen messages will still indicate that **ESC** should be pressed, but the key defined in the VDU parameter file should be used instead.

- 23. **sagerep:** The environment variable **SCPRINTER** now fully over-rides any printer parameter file specified on the command line.
- 24. **sagerep:** The separator character was mistakenly defined as an **a2** type. It is now an **a1** type.

OUTSTANDING KNOWN ERRORS

1. **menu:** Backslash characters are stripped from command line arguments passed to another sub-menu. This is particularly troublesome under DOS. [A temporary workaround is to prefix the menu command with a - (minus) sign, to force the menu program to reload itself and pass the full argument (with backslash characters).]
2. **sage & sagerep:** The use of data dictionary fields with the same name as special temps does not cause an error and may lead to confusion as the field defined in the .d file will be not be used.
3. **sage & sagerep:** The **put** command always translates backslash "\" characters to special characters, such as \t=tab, \n=newline, etc. This prevents the use of single backslash characters in text which will be **put**. If a single character is needed, two backslash characters should be enclosed in the text, ie "\\ ". A future release will have the ability to disaable this translation feature if it is not required.
4. **sage & sagerep:** The **setstr()** function returns the error "Inappropriate argument" when a function (such as **getstr**) is used as it's last argument.
5. **sage:** If a highlighted field in the scroll area is overlaid by a drawbox associated with another screen, the field will be *redrawn* without highlighting.
6. **sage:** When using **editmode 2**, if two fields are placed such that the smaller width field is placed directly below the larger width field then pressing the DOWN-ARROW key near the end of the larger field may result in the cursor being placed outside the input area.
7. **sage:** The format "000,000" is displayed as "0000000".
8. **sage:** Under DOS, the background colours are OR'ed with the standard background colour. This may cause unexpected colours or attributes to be shown if the standard background colour is not \$00 (black).
9. **sage:** The **instr()** function does not correctly deal with subscripted array variables when they are used as the first argument (the string to search).
10. **sage:** The **decdate** command does not deal with subscripted date variables.
11. **sage:** The **at** command is unpredictable if given ridiculous values.
12. **sagerep:** Special temps cannot be redefined in **sagerep**. (Note that this may prevent the use of the **separator** special temp if a separator of more than one character is required.)

- 13. **sql:** In the **where** clause, *logical* expressions may not be evaluated correctly if the expression is particularly complex.
- 14. **sp:** The screen painter allows the title to be erased.
- 15. **sp:** The backslash character is not being output when it is enclosed in curly braces, so using **sp** on a file will strip backslash characters from option help lines and drawbox titles.
- 16. **sp:** Scroll line *headings* that are wider than their fields are truncated on display. The correct heading will be displayed when the program is executed.
- 17. **pdes:** Date validation lists are shown without punctuation.

Sculptor

RELEASE NOTES

Version 2.1

4GL + SQL Application Development System

C O N T E N T S

INTRODUCTION	1
WHAT'S NEW IN THIS RELEASE	2
UPGRADING FROM VERSION 2.0a	5
COMPATABILITY WITH PREVIOUS RELEASES	6
ESCAPE-DETECT MODE	9
MACHINE SPECIFIC INFORMATION	10
UPGRADING FROM VERSION 1.14 or 1.16	12
VERSION NUMBERING	16



INTRODUCTION

Since Sculptor 2 was released, there have been a number of intermediate versions. The initial release, version 2.0, was followed by version 2.0a. This change was concerned with a restriction on the number of open files allowed under the MSDOS operating system. This document details the changes from version 2.0a onwards. The sections are summarised below.

What's new in this release

Everyone should read this section. It details the features that have been added to Sculptor with this release.

Upgrading from 2.0a

Read this section if you are upgrading from version 2.0a or below. It details the new features which were added in previous versions of Sculptor 2.

Compatibility with previous releases

This section specifies any changes which have an impact on the compatibility of existing systems written in Sculptor. Read this section carefully if upgrading an existing system.

Escape-detect mode

This section outlines the input method used by Sculptor and how it may be configured for correct operation on a variety of systems.

Machine specific information

This section details any changes that have been made to deal with individual makes of machines. Check to see if your system is listed and requires special attention.

Upgrading from 1.14 or 1.16

Read this section if you are upgrading from Sculptor version 1.14 or 1.16.

Version numbering

The version number system used by Sculptor is explained in the last section.

Please remember to take a full backup of your Sculptor system before upgrading.

WHAT'S NEW IN THIS RELEASE

Full details on the new features may be found in the Reference Manual.

1. **sage:** [2.1] A new declaration has been added that enables any field to be scrolled to any depth. The declaration syntax is:

```
+scroll fieldname, depth
```

where *fieldname* is the name of a field previously declared on the screen with a "+" field declaration and *depth* is the depth to which the field is to be scrolled. The field may be located at any position on any screen.

Fields declared with **+scroll** are not affected by being placed on the standard scroll line.

2. **sage/sagerep:** [2.1] The following new file commands have been added. These commands operate in the same way as their similarly named counterparts, but they do not attempt to lock the record being read. On most existing operating systems this means that a record may be read even if it is locked by another user (exceptions are DOS Networks and OS9 which do not allow a locked record to be read by another user). The commands are:

readu	read a record
nextu	next record
prevu	previous record
findu	find a record
matchu	next matching record

3. **sage:** [2.1] An additional mode has been added to the **keycode()** function to identify if a key has been pressed *without* waiting. The function **keycode(2)** immediately returns a 1 if there is a keypress waiting or a 0 if not. It does not read the keypress, which remains in the keyboard buffer. Example:

```
while( keycode(2) = 1 ) { /* clear key input buffer */  
    keypress = keycode(0)  
}
```

Once it has been established that characters are waiting, they must be read by the current program. This is to ensure that the program functions in the same way on all operating systems.

4. **sage:** [2.1] User licence information is available using a new function, **users()**. This currently has two integer modes (1 and 2) and two alpha modes (4 and 5) which return:

users(1)	Serial number of Sculptor runtime system
users(2)	Number of licenced users

The above modes return an i4 type.

users(4) Operating system type

This returns an alpha value from the following: MSDOS, UNIX, UNIFLEX, QNX, OS9 or VMS. This function may be used to determine the run-time operating system type in use.

users(5) Operating system sub-type

If the operating system type returned from users(4) is MSDOS, this function returns the operating system sub-type from the following list: NOVELL, MSNET, CDOS and MSDOS. This function returns nothing if a non-DOS system is in use.

5. **sage/sagerep: [2.1]** The function **getenv()** has been added to read data from the environment. Examples:

```
myterm = getenv("TERM")
ushell = getenv("SHELL")
```

6. **sage/sagerep: [2.1]** The function **tostr()** has been added to allow a numeric to be converted to a string. Although Sculptor performs automatic conversion, some situations require that a string argument, not a numeric be used. The syntax is:

```
string = tostr(expression, format)
```

The *expression* is formatted using *format*. If *format* is an empty string, a default is used and leading spaces are removed. The return value is a string. Examples:

```
message "Number is " + tostr(xno, "###")

pfmt = "#,###.###"
astring = "Percentage = " + tostr(anum, pfmt) + "%"
```

7. **cyteol: [2.1]** The **cyteol** program has been added to the suite. This program allows conversion of end-of-line formats in text files. See the manual addendum for further details.

8. **sage/vdu parameter files: [2.1]** On DOS systems each video attribute defined in the vdu parameter file is binary or'ed with the normal background colour. This allows the background to be changed without modifying every attribute in the vdu parameter file. This system is suitable for most colour combinations, however certain mixtures of colours are only available if the background colour is black.

For example, if the normal background is blue (\$01) and an attribute of white on black (\$07) is specified, the result is white on blue. Similarly, if bright white on red (\$4f) is specified, the result is bright white on magenta (\$5f).

Prefixing an attribute byte with \$ff prevents or'ing of the normal background colour for that attribute. In the example above, to obtain bright white on red no matter what normal background is set, specify the attribute \$ff,\$4f. If the standard background is black (\$00) or is blank, the \$ff prefix need not be used.

9. **sage/sagerep:** [2.1] Sculptor allows the use of the standard Unix escape character '\v' on all operating systems. In certain situations this can cause problems because Sculptor recognises this character and treats it specially. An example is reading data with **get** which contains DOS pathnames where the character '\v' is a part of the name. A new special temporary variable, **escchar**, has been introduced to allow the escape character to be changed. **escchar** is an **a1** type and can be set to any printable character. Assigning an empty string to **escchar** disables special character checking. Examples:

```
escchar = ""      /* disables character checking      */
escchar = "!"     /* "!" becomes the escape character */
```

10. **sage/sagerep:** [2.1] To help with the output of "unprintable" characters, such as NULL, a new format code has been introduced which operates in association with the **put** command. A field with the new "b" format causes **put** to output the content of the field as binary characters. All numeric types except **r8** and **m8** can use the new format code. The "b" format is not used by any other command. For example, to output ESC,NULL to a device:

```
i1tmp1 = 27: i1tmp2 = 0
put #1, i1tmp1 format "b"; i1tmp2 format "b";
```

Note that **i2** and **i4** field types are output in natural machine order, not the Sculptor portable order.

11. **vno:** [2.1] The **vno** command returns a revision of 5 for 2.0 compatible and a revision of 6 for 2.1 compatible files (.g, .q).
12. **sp:** [2.1] When a field is selected, the F7 and F8 keys decrease and increase the scroll depth of that field respectively.
13. **describe:** [2.1] A field may now be toggled between key and data using the F5 key.
14. **describe:** [2.1] A new menu option, **Rename**, has been added. This allows the name of the current file to be changed. The **Save** option may then be used to save the file using the new name.
15. **sage/sagerep:** [2.1] A new environment variable **SCNOSHARE** is recognised. Setting this environment variable to any value disables the check for the presence of **SHARE.EXE** on MSNET versions of Sculptor. This option has been added because certain networks do not implement **SHARE** and checking for its existence may cause problems.

UPGRADING FROM VERSION 2.0a

Read this section if you are upgrading from version 2.0a or below. The version numbers in brackets below indicate the release in which the change was made.

ENHANCEMENTS IN VERSION 2.0b

1. **vdu parameter files:** [2.0b] An entry has been added to provide support for terminals which use embedded attributes. Currently this is used by Sculptor only when displaying titles in **sp**, **describe**, **menu** and **sage**.
2. **sagerep:** [2.0b] The **put** command outputs the system end of line sequence. When outputting to channel 0 an additional carriage-return is added to the output stream. This ensures that the cursor is properly placed when the terminal is operating in raw I/O mode.
3. **sage & sagerep:** [2.0b] The **!handles** declaration has been implemented to provide the ability to open up to 32 Sculptor files and 16 sequential files under MSDOS versions 3.3 and above. The command has the following syntax:

!handles *number*

where *number* is the maximum number of DOS file handles to allow in the range 21-255. Each normal Sculptor file requires two handles. Each index only Sculptor file and each sequential file requires one handle.

The total number of file handles cannot exceed that set by the **FILES=** parameter in the **CONFIG.SYS** file. Any attempt to set **!handles** to less than 20, greater than 255 or greater than the **FILES=** setting is ignored.

This declaration should only be used if your program requires more than 8 Sculptor keyed files, or if you are getting an operating system error number 4 (not enough file handles) even after setting **FILES=** to a sufficient number. This declaration is ignored under all other operating systems and under versions of DOS prior to 3.3.

4. **sage:** [2.0b] When writing to a terminal using **put**, the **Ignore protection and disable scroll** sequence (vdu parameter file, entry 19) is issued prior to displaying the characters. The **Honour protection and disable scroll** sequence (vdu parameter file, entry 20) is issued after the characters have been displayed. This allows **put** to place characters over protected fields on the screen.

5. **sql:** [2.0b] The data dictionary field "Hide from SQL" has been implemented according to the following rules:
 - a. In **select *** the hidden fields are not shown and no error is generated.
 - b. A hidden field gives an error number 96 when accessed directly via the **where**, **order by** or **select** clauses.
6. **sage:** [2.0b] Full support has been added for the extended character set between ASCII 128 and 255 for both display and input.
7. **sage:** [2.0b] A new environment variable **SCCOUNT** has been added to control the recognition of special input characters. **SCCOUNT** may be set to a number in the range 1 (the default) to 20. This controls the length of time an input routine will wait to see if a special character is being input.

Please see the section entitled "ESCAPE-DETECT MODE" for further details.
8. **sagerep:** [2.0b] The environment variable **SCPRINTER** fully over-rides any printer parameter file specified on the command line.

COMPATABILITY WITH PREVIOUS RELEASES

This section details changes which may have an impact on existing systems. The version numbers in brackets below indicate the release in which the change was made.

Version 2.1 is fully downward compatible - it will run programs compiled on 2.0b and below. In addition, runtime systems for 2.0 will run programs compiled under 2.1 provided that they do not use any of the features introduced in this version. If a program contains 2.1 specific elements the 2.0 runtime will report:

PROGRAM IS OUT OF DATE

If this occurs the runtime system should be upgraded to 2.1 or the program edited and recompiled, omitting the 2.1 features.

COMPATABILITY WITH VERSION 2.0a

1. **spp:** [2.0b] The **case** statement now falls through to the following **case** statement if a **break** is not encountered. In version 2.0/2.0a only, the **break** was erroneously assumed if a subsequent case was found. Existing programs may require insertion of **break** statements to preserve program flow.
2. **spp:** [2.0b] The special **CONTINUE** statement is recognised as a reserved word. Any occurrence of **CONTINUE** in an existing program will have to be changed if the program is to be compiled with **spp**. See the reference manual for the usage of this statement.
3. **menu:** [2.0b] The **ESCAPE** key exits the menu and does not select the current option.
4. **menu:** [2.0b] The **\$** sign is now retained within the command line and is passed on to the called program.
5. **sage:** [2.0b] The keyboard interrupt handling under DOS systems has been improved and it is now possible to process a keyboard interrupt at any time, not just when at an input field.

6. **sage:** [2.0b] When the **lock** command is used on a file, any attempt to insert a record on the file by another user will cause that users program to wait until the lock is released. If the process which placed the file lock issues an **insert** or **delete** command, the file lock is removed before the command is executed. The file lock is not re-instated. A **write** command will remove the file lock if the key values have been changed since this is equivalent to an **insert** and a **delete**.
7. **sage:** [2.0b] When clearing the prompt line, only the text itself is cleared, not the whole line. Programs which write to the prompt line and assume it has been cleared will require some modification.
8. **sage:** [2.0b] The **put #0** command uses the current video attribute under DOS.
9. **sage:** [2.0b] When using redirected input, the input stream is read character by character rather than using buffering. This allows **exec** commands within that program to use the redirected input stream correctly. This applies only to Unix systems.
10. **sage:** [2.0b] The block option line now recognises the SPACEBAR as being equivalent to cursor right and does not execute the highlighted option.
11. **sagerep:** [2.0b] The separator character is now an **a1** type.

COMPATABILITY WITH VERSION 2.0b

12. **sage/sagerep:** [2.1] The **rand()** function returns an **i4** type instead of an **r8**.
13. **spp:** [2.1] Errors reported from the pre-processor now show the line from the original source file and not the intermediate output file. A new option, **-e**, may be used to force **spp** to show error lines from the intermediate file.
14. **installation:** [2.1] When installing Sculptor, the user is now asked for a serial number and an activation key.
15. **sage:** [2.1] If there is any entry in the vdu parameter file which is set to a single **ESC** character, Sculptor automatically sets **ESCAPE-DETECT MODE** and uses the value of the environment variable **SCCOUNT** to determine input timing. If a single **ESC** is not found then **ESCAPE-DETECT MODE** is disabled and the value of **SCCOUNT** is ignored. See the section entitled **ESCAPE-DETECT MODE** for further details.

ESCAPE-DETECT MODE

In order to make SCULPTOR portable across a wide range of operating systems, the ESCAPE character can be detected as a single keypress as well as a lead-in to a full escape sequence. This is known as **ESCAPE-DETECT** mode.

Most vdu parameter files are shipped with the END OF INPUT entry set to ESC. This enables ESCAPE-DETECT mode. Some machines, notably those with intelligent i/o controllers, are confused by the use of the ESCAPE key. The problem is typified by the unexpected triggering of the END OF INPUT trap or by parts of an ESCAPE sequence being displayed. There are two ways to overcome this problem.

a) Set the environment variable SCCOUNT to a number in the range 1-20. A value of 8 is normally adequate. This has the effect of slowing the response to the ESCAPE key. If this is not acceptable the ESCAPE key cannot be used. The default value of SCCOUNT is 1.

b) Set END OF INPUT to an unused key such as END (on a PC keyboard). Ensure there are no other single occurrences of ESC in the vdu parameter file. This maps the ESC function to whatever the END OF INPUT key is set to.

On most systems this environment variable should not be used. Some terminals, however, send their function key sequences very slowly. Increasing the SCCOUNT value allows special character strings to be recognised as a single key press.

IMPORTANT NOTE: When ESCAPE-DETECT mode is disabled the development system programs, **sp** and **describe**, still indicate that ESC can be pressed, but the key defined in the vdu parameter file should be used instead.

Under 2.1, SCCOUNT is automatically set to zero (ESCAPE-DETECT mode disabled) if a single ESC is not found in the vdu parameter file. This did not take place in previous versions of Sculptor.

SCCOUNT with a value of zero speeds up the entry of escape sequences. Unidentifiable keys however (i.e. those not present in the vdu parameter file) cannot be detected. If pressed, such keys may cause the program to stop and await further input.

MACHINE SPECIFIC INFORMATION

PRIME EXL

This machine appears to have a problem interpreting the ESCAPE key. If, under the normal configuration of Sculptor, ESC is pressed the program will hang. This fault may be specific to certain machines. If your Prime exhibits this problem take the following action:

- a) Replace the entry for 'End of Input' with a value other than ESC, for example the Home key.
- b) Set the environment variable SCCOUNT to zero.

See the section entitled "ESCAPE-DETECT MODE" for further details.

SEQUENT BALANCE

It has been noted that on some configurations of the Sequent Balance, those that use PC's as terminal emulators, the ESCAPE key is not passed transparently through the network. If this is the case, the ESCAPE key cannot be used.

SUN

There are two vdu parameter files for Sun machines. One for the 386 and one for the model 3 machines. The appropriate one should be copied to the name given by \$TERM. The main difference between the two files is the keyboard codes.

When running under Sun View the standard emulation does not provide thin line graphics characters. If you require these graphics for better screen presentation you should run the 'vt100tool' available from Sun. In addition X-windows does not support the use of ESCAPE as a function key. The "end of input" sequence has been changed to ^A. See section entitled ESCAPE DETECT MODE.

QNX

Sculptor requires release 3.15B or greater of QNX.

IBM RS/6000

The vdu parameter file "ibm3151" operates this terminal in native mode only. The vdu parameter file "aixterm" will operate under X-windows on the 6091 terminal. Other X terminals have not been tested at this time.

WYSE TERMINALS

In the wyse50 and wyse350 files, the 'Del' key is now used for backspace so that the left arrow key (which sends a backspace code) can be used to move the cursor left without erasing characters. The keyboard interrupt is set to Ctrl C. To delete the character under the cursor, use the 'DEL Char' key.

On terminals which require a space to set an attribute (e.g. wyse50), attributes can only be used where the area to be highlighted can be pushed one character right without damaging the display. This means that the title line (maximum 78 characters) can have an attribute but field headings and data fields cannot.

This restriction does not normally apply to the dim attribute which is commonly used on protected fields. Because of the limitation on setting attributes, it is difficult to make good use of colour on the wyse350. The parameter file supplied provides a reasonable screen on a wyse350 which is correctly set up.

UPGRADING FROM SCULPTOR VERSION 1.14 OR 1.16

1. Version 2.1 has been designed to be compatible with previous releases (version 1.16 and 2.0). The procedure for upgrading is detailed below. Please ensure that you have a backup of your system before proceeding.
 - ❖ *Before you can start the upgrade, you must already have installed version 2.1 on your system (see the installation notes supplied). If you have not done that, please do it now.*
2. When version 2.1 has been installed, ensure that your PATH environment variable is correctly set to locate the version 2.1 programs. Typing `sage` at the command prompt will give you the header information for the `sage` program. Check that version 2.1 is being executed. If it is not, set the PATH variable accordingly (see the installation notes).

UPGRADING FROM VERSION 1.14 OR EARLIER

3. Before you can upgrade a version 1.14 or earlier version of Sculptor to version 2.1, you must first upgrade your data files to the version 1.16 format. Between 1.14 and 1.16 changes were made to the way in which SCULPTOR stores data and to the way the intermediate code files were created. This was done to enable programs to be run unchanged on a variety of different operating systems. The procedure is as follows:
 - 3.1. **TAKE A FULL BACKUP.** This is important; do it now.
 - 3.2. All data and index files must be converted using the `cvt1416` program. Please refer to the attached documentation for `cvt1416` for a complete description of its operation. In simple terms, though, it involves little more than typing,

`cvt1416 directoryname`
 - 3.3. For each directory which contains data files to be converted, this performs the necessary conversion to the new format. Please read the `cvt1416` documentation and then convert your data files now.
 - 3.4. All vdu and printer parameter files must be decoded using 1.14 `decprinter/decvdu` commands. In the decoded files, make the additional entries required for Sculptor 2.1 and recode using the 2.1 versions of `setprinter/setvdu`.

- 3.5. You should now perform the upgrade from version 1.16 to version 2.1 detailed below.

UPGRADING FROM VERSION 1.16

4. Six factors must be considered before upgrading version 1.16 to version 2.1:
- 4.1. **SCULPTOR** now has a well defined directory structure, with sub-directories holding programs and parameter files. Care should be taken to avoid installing over an existing directory. The `'etc/sage'` directory is now redundant. On UNIX and DOS systems the environment variable `PATH` must be updated to include the program directory `'SCULPTOR/bin'` (see the installation notes).
 - 4.2. The infield editing feature (used by the `'input'` command in the screen form language) will be unfamiliar to users of version 1.16. Many elements of the editing may be configured.
 - 4.3. The editing keys must be correctly configured for in-field editing to work (see the VDU parameter file documentation). The only visual effect of using programs compiled with version 2 is the `INSERT/OVERTYPE` indicator that is displayed on the message line.
 - 4.4. The vdu and printer parameter files have been substantially modified. If a vdu or printer parameter file for your system has not been supplied, you will have to convert your existing parameter files as shown below.

The new versions of the parameter file utilities can be used to convert old(1.16) files to new. They add empty entries for the extra lines and add new header information.

The following steps must be used to produce a Version 2 parameter file from a v1.16 parameter file. (This is an example for a VDU file, but the same principle applies when using `decprint` and `setprint`) :-

- i. Make a copy of the parameter file for safety.
- ii. Enter `decvdu -cf wy50`. This creates a file called `'wy50.s'` in a modified 1.16 format.
- iii. Enter `setvdu -cf wy50` this remakes the file `'wy50'` using the 2.1 format.
- iv. Delete the file `'wy50.s'`. This file is only partially converted and should not be used.

- 4.5. This new file can be modified in the following way to include the new codes needed to make **SCULPTOR** run with your terminal.
- i. Enter `decvdu -f wy50` to create a new 'wy50.s' file.
 - ii. Edit this file with your text editor.
 - iii. Enter `setvdu -f wy50` to remake the parameter file.
- 4.6. The interrupt key. Users of Unix systems are now able to define which key will perform the keyboard interrupt function. This is set in the vdu parameter file and as a convention is ^C (Control - C). If your interrupt key stops working when you install version 2.1 it may have been set to something else. Edit the decoded vdu parameter file accordingly.
5. Four minor changes have been made to the SQL package.

5.1. The configuration file

Now that the vdu's and parameter files have been split into separate directories there is no obvious place to locate the sql configuration file. Therefore a directory called **default** has been created within the **Sculptor** directory and any existing configuration files must be moved there.

5.2. Date styles.

The date style control in the configuration file has been changed to bring it into line with the standard **Sculptor** date format. A typical line from the configuration file would now read

`DATE = "dd/mm/yy"`

5.3. Numeric Formats.

An error that prevented field formats being read from the data description file has been corrected.

This may mean that existing queries which do not use a 'format' clause may display data slightly differently.

5.4. Dimensioned Fields.

It is now possible to total dimensioned fields in the 'select' clause:-

```
select value[1 to 5]  
select value[all]
```

These two examples show how to sum a range of elements in a field and sum all the elements of a field respectively.

6. A number of other factors must be taken into account when upgrading an existing application to version 2.1 from either 1.16 or 1.14. These are:

6.1. Reserved words

Sculptor now has over 150 reserved words. These words cannot be used within screen form or report language programs. A list of the reserved words can be found in the Reference Manual. Additional reserved words in the 2.1 release are:

tostr	users	getenv	escchar
toupper	tolower	readu	nextu
prevu	findu	matchu	users
informix	sqlexec	sqlkey	

6.2. Program name changes

The utilities **setprinter** and **decprinter** have been renamed **setprint** and **decprint**. Please ensure you do not use the old versions of these programs inadvertently.

6.3. File limits

The overall number of **Sculptor** keyed files that a single program can have open has been increased to 32. Under DOS this requires the use of the special declaration **!handles**. See "Upgrading from version 2.0a" for details.

6.4. New directory structure

As the **Sculptor** directory structure has undergone substantial changes, applications that use the old directory structure explicitly will have to be altered.

6.5. Nested gosubs

Due to the slightly increased memory overheads for the interpreters **sage** and **sagerep**, the limit of 100 nested **gosubs** has been decreased to 95.

6.6. Environment variables

Sculptor uses a number of new environment variables and no longer uses the **ETCSAGE** variable. See the Reference Manual or the Pocket Reference Guide for details.

VERSION NUMBERING

1. Each program in the SCULPTOR package has a version number. In general, a program that is run without a command line argument displays its identification. Those which normally run without a command line argument, display a message box containing their identification when the programs terminates.

A version number comprises four sections as follows:

2 . 1 a (10 August 1991)

=> |a| b |c| d |

- a. Major release number, in this example 2.

This changes when major new features are introduced in SCULPTOR.

- b. Minor release number, in this example 1.

This changes when minor enhancements are made to SCULPTOR.

- c. Release revision letter, in this example a.

Corrections required between releases are denoted by appending a lower case letter.

- d. Reference date.

The date is for use by MPD. It changes when internal modifications are made for a particular machine or operating system.

Any request for technical support *must* quote the full version number together with the serial number of the product.

Sculptor

Sculptor version 2.4

Release Notes

© Sculptor Ltd 1994

Keyed File Libraries for 'C' users read notes on the last page.

Published by

Sculptor Ltd

Sculptor 2.4 Release Notes

General

Version 2.4 release of Sculptor incorporates version 2.3 and contains the following major items:-

- Transaction Logging
- Alternate Indexing
- Extended Sculptor Language
- New Sculptor Language Compiler
- New Format VDU Files
- New Utilities cvtvdu, describe, kfbuild and scview
- International Language Facility

All of these items are covered in detail in the following sections.

Transaction Logging

Version 2.4 introduces a transaction logging and recovery system. The transaction logging system can be used in two ways:-

1. As a method of **recovering a corrupted database** following a system crash or other similar catastrophic event. In this case the database is restored in the normal way from its last backup (usually on tape) and the recovery program uses the logged transactions to restore the database to the point of the last completed transaction.
2. As a method of **mirroring a database**. In this case the recovery program is run continuously, reading the logged transactions and adding them to a fallback database as each transaction is completed.

It is not necessary to modify or recompile existing programs to begin using transaction logging, because each of the database changes are treated as individual transactions. The full power of the system is realised when the new Sculptor commands **begin** and **commit** are inserted into programs to group together a number of database writes to form a single transaction. A further Sculptor command allows for logging to be switched on and off under program control.

Transaction logging is controlled by a Sculptor **screen form** program which has options to initialise transaction logging and to control the **recovery program**. There is also a **batch control program** which allows the recovery system to be controlled from a shell script.

Once transaction logging has been started, all updates to all Sculptor keyed files are written to a common log file as well as the normal Sculptor files. The log file is a sequentially written file, not a keyed file. The Sculptor utility programs (**kfcopy**, **reformat** and **newkf**, *but not **kfri***) also write to the log file. Writes to non-Sculptor files are not recorded in the log file.

The transaction logging and recovery system is fully described in the associated documentation.

Alternate Indexing

Sculptor Version 2.4 introduces the concept of an Alternate Index.

In the past a Sculptor keyed file has been used as a primary index into a data file. If the Sculptor programmer wished to have indices on other fields then this was achieved by creating index-only files which incorporated the required alternate index key and the unique key from the master Sculptor keyed file. This system works adequately, but requires that the user perform a number of operations (one per key file) to insert, delete or modify a record.

The **Alternate Index** facility is introduced with Sculptor version 2.4, which allows the user to specify a number of alternate indices which are automatically and simultaneously updated when a user performs an insert, delete or write (modify) operation on a data file. A new **index=** clause is introduced for file access commands which allows for an appropriate alternate index to be selected instead of the main index.

A new version of the **describe** utility is provided to allow the specification of the Alternate Index fields. The **describe** utility has been enhanced to provide a more user-friendly interface and utilises the new format VDU files (see below for details). The new Sculptor compiler **scc** generates code to update all the alternate indices whenever a file operation is performed and processes the **index=** clause in all file access commands. The **index=** clause is permitted in a **!file** command in **sagerep**, thereby allowing an alternate index to be the driving file for a generated report.

Extended Language Features

The Sculptor language has been enhanced in the following areas in version 2.4:-

- The **logging**, **begin** and **commit** commands have been added to provide transaction logging facilities.
- The **index=** clause has been added to all file access commands and to the **!file** declaration.
- The **if ... then ... else** command has been enhanced to allow a group of Sculptor commands to be specified following a **then** or **else** instead of a single line. The commands are grouped together using the braces { and }. The old format **if** statement is still compiled correctly.
- A new **breakif** command has been introduced which is only permitted within the braces of the enhanced **if** command. The **breakif** command transfers control to the next statement following the **if ... then ... else** command. A corresponding **trap = BREAKIF** clause is permitted as a special label within the braces of the enhanced **if** command.
- A new **do { ... } while (condition)** construct has been introduced which works in the same way as a **while** command, but ensures that the group of statements enclosed by the braces are executed at least once, irrespective of the *condition* specified.
- File aliases may now be used in **sagerep** source files in addition to file numbers.

Language Compiler

A new Sculptor language compiler **scc** is released with version 2.4. The compiler **scc** now includes all the preprocessing capabilities of **spp** in addition to support for the new language enhancements and for alternate indexing. Hence **spp**, **cf** and **cr** are no longer required and should not be used with version 2.4.

Changed Utilities

There are several changes to existing utilities with Sculptor 2.4:-

- The **kfcheck** utility now has a **-x** option which forces the checking of all the alternate indices. A new **-h** option provides help on the alternative command line options. A **-iname** option allows for only alternate index **name** to be checked. The level of the index check may be set by the **-ln** option (where **l** is a lower case L character and **n** is a single digit number in the range 1-5). Level **n** has meaning as follows:
 - 1 Check only the main index (the default level).
 - 2 Perform level 1 check and read the data file to check key file index content.
 - 3 Perform level 2 check and check keyed file integrity of alternate indexed files.
 - 4 Perform level 3 check and perform 1-pass **main=>index** data integrity check.
 - 5 Perform level 4 check and perform 2-pass **main=>index=>main** data integrity check.
- The **kfri** utility now has a **-x** option which forces the rebuild of all the alternate indices in addition to the main index. A new **-h** option provides help on the alternative command line options. A new **-q** option is added which is identical to the existing **-s** option and makes **kfri** operate in quiet (silent) mode. A new **-iname** option allows for only alternate index **name** to be rebuilt.
- The **newkf** utility will create all the alternate indexes in addition to the main index. The alternate indexes are stored in a directory **filename.x** (where **filename** is the parameter supplied to **newkf**). If this directory does not exist then **newkf** will create it.
- The **pdes** utility without a parameter list provides help on the alternative command line options.
- The **reformat** utility now has a **-u** option which forces the rebuild of all the alternate indices. A new **-h** option provides help on the alternative command line options.
- **eqf**, **fql**, **sp** and **sql** have all been updated to accept the new format file produced by the **describe** program. These utilities do not need to access any alternate indices, but they will accept a file which contains alternate indices.

New Format VDU Files

In Sculptor 2.4 the new format VDU files have been introduced to provide enhanced facilities. The changes have been necessary in order to add functionality and to remove ambiguities. The new VDU files are used by the **describe** and **scview** programs released with Sculptor 2.4 and will be used by other Sculptor software in the future.

At present the new VDU files are used by some Sculptor software, but the old VDU files must be retained for the foreseeable future. Version 2.4 includes new format VDU files for all popular terminals.

The Sculptor version 2.4 VDU files are kept in source form in a file of name <terminal>.s and are converted to a binary VDU file of name <terminal>.v by the new **compvdu** utility. The new VDU files contain a large amount of descriptive information which does not appear in the corresponding binary VDU file. There is no decode utility available for the new format VDU files. The new format VDU files are far more elaborate than the old ones and hence there is no direct conversion available from the old to the new format files.

The **cvtvdu** utility is provided to help developers who are moving to Sculptor version 2.4 and who have terminals for which a standard VDU file is not available.

New Utilities

Three new utilities are released with Sculptor 2.4:-

- **cvtvdu** is provided to help developers who are moving to Sculptor version 2.4 and who have terminals for which a standard VDU file is not available. It creates a new format VDU source file from an old format binary VDU file.
- **describe** is a completely new, user-friendly, menu oriented version of the old data dictionary editor of the same name. All features of the old **describe** are maintained and support for alternate indices added.
- **kfbuid** is provided to rebuild all of the alternate indices from the datafile. A **-iname** option allows for the specification of a single alternate index **name**. A **-h** option provides help on the alternative command line options. Note that **kfri** is still available to rebuild the main index.
- **scview** allows the user to view a text file. A user-friendly interface is provided. The **scview** utility uses the new format VDU files.

International Language Facility

The international language facility is provided so that the messages output by utilities may be changed to a language other than English. The utilities **describe**, **kfbuild**, **kfcheck**, **kfri**, **newkf**, **pdes**, **reformat** **scview** and **scc** all use the international language facility.

The messages are contained in files with the extension **.eng** in the directory **\$SCULPTOR/lang**. The environment variable **SCLANGUAGE** is set to the three character extension of the message files to be used by these utilities. If **SCLANGUAGE** is not set then **.eng** is assumed.

It is the responsibility of the developer to create the corresponding language files by translating the English messages into the appropriate language.

Additional options for **sg** & **rg**

sg and **rg** have been modified to make full use of the automatic alternate indices introduced in Sculptor version 2.4 (the **-a** option will continue to run the **sg** and **rg** programs interactively).

A **Help** option has been included to both **sg** and **rg**, to display the new options available.

The following is a list of command line options now available to **sg**:-

Command line syntax: **sg [-a] filename [options ...]**

-h	Displays this Help screen
-w	Set screen width to 132 columns
-b (delimiters)	Set box delimiters
-s	Form highlight on
-f	Field highlight on
-d	Vertical option box
"-l (description)"	Set user defined logo
-m	Set automatic input help messages

The changes made to **sg** enable it to operate with nine screens full of data and with the first five alternate indices.

The ability to specify highlighting, widths and box delimiters is now configurable using the **-a** option, or by setting the parameters in the default **sgen.cfg** file without any necessity for editing and recompiling the code that has been generated.

A vertical bar menu has been included to facilitate prototyping.

Command line options that are available to the **rg** program are:-

Command line syntax: **rg** [-a] *filename* [*options ...*]

-h	Displays this Help screen
-l (index)	Name of alternate index to use
-s	Suppress duplicate keys

The **rg** program include will now run from an alternate index file that has been defined using the new **describe** program.

IMPORTANT - KEYED FILE LIBRARY USERS SHOULD READ THESE NOTES

Transaction logging and recovery, introduced in Sculptor version 2.3, is applied at the Sculptor keyed file library level. This means that keyed file library users will need to obtain an upgrade in order to use the new features. Transaction logging does not affect Sculptor data files, it is only if a recovery becomes necessary that Sculptor data files are updated.

The Transaction logging and recovery system is initialised and controlled from a **sage** or **sagerep** program supplied with version 2.3 runtime systems. Therefore a version 2.3 run-time system must be in place to use transaction logging from a 'C' program using the keyed file library.

Sculptor version 2.4 introduces a collection of new features into the Sculptor 4GL development environment. One of these is automatic cross reference files. As this is implemented by the Sculptor pre-processor, there are no implications for 'C' programs which use the keyed file library.

The automatic cross reference files are standard Sculptor keyed files which may be accessed from the keyed file library. We strongly recommend that these files are treated as *read only*.

Sculptor Ltd