sun®
microsystems

Software Technical Bulletin

March 1987

Software Information Services

Part Number 812-8701-02
Issue 1987-2
2 March 1987

Software Technical Bulletins are distributed to customers with software/hardware or software only support contracts. Send comments or corrections to "Software Technical Bulletins" at Sun Microsystems, Inc., 2550 Garcia Ave., M/S 2-34, Mountain View, CA 94043 or by electronic mail to sun!stb-editor. Customers who have technical questions about topics in the Bulletin should call Sun Customer Software Services AnswerLine at
800 USA-4-SUN.

UNIX is a trademark of AT&T Bell Laboratories. DEC, DNA, VAX, VMS, VT100, and Ultrix are registered trademarks of Digital Equipment Corporation.
Sun-2, Sun-2/xxx, Deskside, SunStation, SunCore, SunView, SunWindows, and DVMA are trademarks of Sun Microsystems, Inc.
Sun-3, Sun-2, DVMA, SunWindows, NFS, SunView, SunGKS, SunCGI, SunGuide, SunLink, Sun Microsystems, and the Sun logo are trademarks of Sun Microsystems, Inc.
Sun Workstation and SunCore are registered trademarks of Sun Microsystems, Inc.

Copyright © 1987 by Sun Microsystems.

This publication is protected by Federal Copyright Law, with all rights reserved. No part of this publication may be reproduced, stored in a retrieval system, translated, transcribed, or transmitted, in any form, or by any means manual, electric, electronic, electro-magnetic, mechanical, chemical, optical, or otherwise, without prior explicit written permission from Sun Microsystems.

Contents

1

NOTES & COMMENTS

1

NOTES & COMMENTS

As this is being typed, feedback from the first issue of the Software Technical Bulletin, or the STB as we fondly call it, has started coming in. Overall, the comments are very positive and we in Software Information Services (SIS) are very glad you liked it. (We also let out a collective sigh of relief).

We are also paying close attention to the changes you are suggesting, and we are continually looking for new things to try.

For example, in this issue we have published information about the upcoming Sun Operating System release. This information is derived from the 'Read This First' document that ships with all new machines. The STB will periodically provide such information.

Another feature we are planning now is an 'answer column' where selected questions you provide will be explained in detail. More on that in the next issue.

We have also started the redesigning the format for the Customer Distributed BugsList. Published quarterly, look for our new design in the May issue.

Finally, SIS is working very hard to expand our numbers from the current two to three (and beyond!). As the group grows, we hope to bring you bigger and better issues of the STB that discuss what you want to hear. So please keep those comments coming to *sun!stb-editor*. We cannot promise to answer each and every one, but we do promise to read them all and pay attention to what you say. Thanks.

The Editor

2

ARTICLES

2

ARTICLES

Pixrects/Pixwins

Definitions

A pixrect is a rectangular area defined by the user whose purpose is to provide a device independent way of rendering dots, known as pixels, on the screen. The user may use operations such as vectors, text, and raster-ops to render the pixrect image. The pixwin extends the use of pixrects into the window environment, by accomodating for overlap, resize, and movement operations. The pixrect is the element of the pixwin which contains the image to be displayed.

There are basically two kinds of pixrects - screen and memory. A screen pixrect is the entire frame buffer and has the size of that frame buffer. In addition to height and width, a pixrect has depth. In general there are two depths used - 1 bit and 8 bit. 1 bit for monochrome, 8 bits for color. A screen pixrect for /dev/bwtwo0 is 1152 pixels wide by 900 pixels high. A screen pixrect for /dev/cgtwo0 and /dev/cgfour0 is 1152 pixels wide by 900 pixels high by 8-bits deep. An optional memory pixrect is just a chunk of memory of whatever size you want. The pixrect application creates this memory pixrect for copying operations.

Pixwins

A pixwin can be thought of as an intelligent pixrect that knows how to interact properly with a dynamically changing environment. Four extensions are necessary for windows:

1. movement
2. resizing
3. clipping overlapping regions
4. handling damage

Pixwins use pixrects to draw on the screen, and optionally have a memory pixrect as a back-up copy.

The pixwin contains the pixrect structure(s), and provides for their handling in the window environment. The pixwin may be moved, resized, and overlapped by other windows, without damage to the pixwin's contents. When an underlying pixwin is exposed, it's contents are redisplayed. The retained pixrect is used to 'fix' damage that occurs when a window has another window overlap it, or the window is opened, or for redisplay. A perfect example of this is done within the Suntools environment. Think of two shelltools where one overlaps the other. When the underlying shelltool is exposed, it's contents reappear correctly. This is true for all Sun applications within Suntools.

There are two kinds of clipping - one to clip to the rectagular boundaries of the pixwin, and one to clip out any rectangular regions which should not be visible due to overlapping by other windows.

Imagine the case when one draws a vector onto a pixwin.

> Any segment of the vector which goes outside the pixwin rectangular boundary is clipped.
>
> The vector is then drawn onto the retained memory pixrect (if present).
>
> Then the vector is again clipped against any overlapping windows. This removes any parts of the vector which should be hidden by a window which is in front of the window being drawn to.
>
> This vector is then drawn onto the screen pixrect.

Within Sunview, an example usage is to draw within canvases without affecting the display of the surrounding and overlaying windows. Images can be drawn to the canvas using either Sunview, Suncore, or SunCGI routines within the Sunview application. All of the above listed packages must reference the canvas subwindow through the canvas' pixwin. For further information, please consult the 3.0 or 3.2 Sunview Programmer's Guide, Chapter 7 - 'Imaging Facilities: Pixwins', SunCGI Reference Manual, Appendix F - 'Using SunCGI and Pixwins', and the Suncore Reference Manual, Appendix B - 'Suncore View Surfaces'.

Sun Operating System Release 3.3

During February, Sun released the 3.3 upgrade to its operating system software. The 3.3 upgrade includes the following:

- SCSI disconnect/reconnect feature enabled for Sun-3/52 models - i.e., Sun-3/50 with the following peripherals:

Shoebox Disk	Option Numbers
71 MByte (Adaptec Cntlr)	501 A, B, or D 503 A 511 A, B, or D
141 MByte (Emulex Cntlr)	504 A, B, or D 505 A 514 A, B, or D

Hardware: For 3/50 and 3/52 systems, this capability requires an MT02 tape controller and the on-board SCSI.

Unsupported Configuration: The Release 3.3 disconnect/ reconnect implementation will not support 3/50 or 3/52 systems with a 141MB disk as the first disk on a daisy chain and a 71MB disk as the second disk on a daisy chain. (This restriction will be removed by Release 3.4.)

- Limited support for Internet standard subnets.
- The tbl bug affecting tables columns specified as n has been fixed.
- Two bugs affecting NFS file locking which were reported in the 3.2 *Read This First* document are fixed.
- Enhancements to the Pascal compiler.

Everything in the 3.3 upgrade will be incorporated in Sun's 3.4 software release.

Since the 3.4 release is scheduled to ship starting in April, 1987, Sun will ship the 3.3 release to customers with support contracts who request the release through the Sun AnswerLine at (800) USA-4-SUN. If one of the bugs mentioned above does affect your use of a Sun product, and you cannot wait until release 3.4 is shipped, please call the AnswerLine and ask for a copy of revision 3.3. You must be ready to supply your Sun support contract information.

Regardless of your getting 3.3 or not, you will receive the 3.3 software features as part of your 3.4 upgrade sometime after mid-April.

SunLink Patches Available

This is a list of the SunLink patches available for customers, from Custsomer Software Services. If you see a problem description below that you know you are experiencing, please contact the Sun AnswerLine.

HASP (RJEBSC)

1. Correct handling of the wait-a-bit bit. Under certain conditions we did not correctly clear the wait-a-bit bit and would never resume sending. Symptoms would be that jobs get stuck in the transmit queue and are never sent even though host appears up. This fix only affects HASP mode.

2. New version does not send a 'request to initiate' on the console command path when sending operator commands. This bugs causes trouble on Univac systems when two operator commands are sent quickly or an operator is queued to be sent when the rjebsc process is started. This fix only affects HASP mode. Workaround: remove SRJEQ file when starting rjebsc program and do not send two operator commands in quick succession.

3. Core dump that occurs sometimes after the message

```
'CONSOLE CHANGED TO....'
```

This fix only affects HASP mode.

4. If rjebsc is run on a very busy Sun system, the host and the Sun may get out of sync. If you see occasional rjebsc failures on a busy Sun running rjebsc in HASP mode, you might want to try this version. WORKAROUND: run rjebsc on less loaded machine.

For any of the above ask for patch:

```
bscrje/hasp_fixes
```

SNA3270

A bug fix to SunLink SNA3270, Release 3.0. This fix should solve the problem with SNA3270 displaying the following error:

```
panic 10
```

Ask For

```
sna3270/panic_10_fix
```

SNA3270

A bug fix to SunLink SNA3270, Release 3.0. This fix should allow the ibmftp program to transfer files to an IBM host running VM and MVS.

Ask For

```
sna3270/ibmftp_fix
```

SNA3270

More than 4 SNA3274 processes cannot be started on the same sna gateway. When you try to start the 5th cluster controller process, you get:

```
<time stamp> 3270 server: register with mapper failed
```

This was introduced in SunLink 3.0 SNA 3270 and carried forward in SunLink 4.0 SNA 3270. The 3270mapper has only 8 slots for registering instances of the gateways. In 3.0 and 4.0, each SNA 3270 gateway registers itself twice: once for version 1 of the protocol and once for version 2 of the protocol. Since there are only 8 slots, this means that a maximum of 4 gateways could be started on a single machine.

This will be fixed in SunLink 5.0 SNA 3270 so that up to 8 gateways can be started on a single machine.

Ask For

```
sna3270/more_than_4_gw_fix
```

SCP

There is a patch that contains fixes to release 3.0 SCP software. The bug fixed by this file is the algolrithm for allocating buffers. A group of buffers are allocated for usage instead of on an as needed basis.

Ask For

`scp/68020` or `scp/68010`

X.25

Bug fix to SunLink X.25, Release 4.0. This fix should solve the problem where when a RESET INDICATION arrives on a busy system where packets are queued for transmission, Sunlink X.25 may not recover properly after transmitting the RESET CONFIRMATION. Consequently higher level protocols layered above X.25 may lose their connections.

Ask For

```
x25/reset_fix
```

BSC3270

In release sunlink3.0, users trying to send output to printers that they have defined in the BSC 3270 environment will not succeed. pe3287 dies with a panic code of 3 because its buffer size is too small.

Ask For

```
bsc3270/panic_3
```

Data Communications/Networking

Connecting Sun and DEC Equipment

This memo provides information on the different methods available for connecting Sun and DEC (Digital Equipment Corporation) computers together.

Asynchronous Connection - Terminal to Host Communication

Asynchronous terminals can be connected to DEC computers. Sun workstations can emulate asynchronous terminals through either local ports or ports on the ALM board. The Sun and DEC computers can be connected together by using an RS232C interface and asynchronous communication.

`Shelltool` in SunTools does not emulate a standard terminal. By using the SunLink VT100 product, the Sun will emulate a VT100 terminal in SunTools.

□ Advantages

Low cost connection, simple to understand, implement and use.

If a Sun is connected to a DEC computer, any Sun or PC (via remote login) in the Sun internetwork can make use of the connection.

Can be used with any DEC operating system or type of computer.

□ Disadvantages

Speed is limited to 9600 or 19200 bps (bits per second)

A separate port on the Sun and DEC computer, plus a cable, is needed for each connection to the DEC.

File transfer is slow and difficult to use. The file transfer capability is ASCII only. Binary files must be converted to and from ASCII before they can be transferred. No automatic file format conversion between UNIX file format and DEC O/S file format is performed. Use the Kermit program to accomplish asynchronous file transfer between a Sun and a DEC computer.

Sun DECNet Emulation - Host to Host Communication

By using the SunLink DECNet product, a Sun workstation can connect over an Ethernet to a DEC machine that is running DEC's DECNet networking software. The Sun can use remote login to the DEC computer and vice-versa. The Sun can store, list and retrieve files on the DEC computer. Finally, a programmatic interface is available on the Sun that allows a Sun program to communicate over the DECNet session level software to a cooperating program on the DEC computer.

- Advantages

  No change is needed to the DEC computer. This is a big win since the people who control the customer's DEC system will prefer (or often require) that no changes are made to their computer. The SunLink DECNet product means that the Sun user can interoperate with the DEC computer system with absolute minimum effort from the DEC system manager.

  The connection is made by using software and hardware supplied by DEC and Sun as standard products. No third company products are needed.

  The connection between the Sun and the DEC computers is over the Ethernet. The product provides high speed, multiple sessions between Sun and DEC computers.

  No extra hardware is needed on the Sun.

  Any DEC computer, and any operating system with DECNet Phase IV can be accessed.

  The SunLink DECNet product includes the 'SunLink VT100 Terminal Emulator'.

  PC's, using Sun's PC-NFS can access the DEC network via a Sun.

- Disadvantages

  The Sun can log on to the DEC computer and access DEC mail, but the product does not include a mail gateway between the DEC and Sun mail systems yet.

  the SunLink 4.0 DNA-Type Protocols product cannout use remote login to DEC Ultrix systems. This restriction will be removed in the future.

DEC Ultrix TCP/IP - Host to Host Communication

The DEC Ultrix operating system is a port of Berkeley Unix 4.2. As a result, Ultrix includes support for TCP/IP and the Berkeley networking utilities. If an Ethernet interface is installed, then DEC and Sun computers can communicate over the Ethernet using FTP, Telnet, Mail, rlogin, rsh, etc.

- Advantages

  No extra software is needed on either the Sun or DEC computers.

  A Sun/DEC solution; third party products not used.

3

IN DEPTH

3

IN DEPTH

Sun Common Lisp Version 2

Sun Common Lisp Version 2 offers significant enhancements to the previous Lisp system Sun has released. These begin with the process of installing the Lisp system and carry through to new programming features and much better documentation.

New Features

SCLISP Version 2 has three new programming components. These are:

□ a Window Tool Kit

□ a built-in Emacs like Editor

□ implementation of Flavors.

All three are optional. None of the kernel Lisp system depends on these additions. At the time Lisp is installed, the site can choose to use any combination of the Window Tool Kit, Editor and Flavors.

Other Problems Fixed

Besides adressing the issue of optional components, the new installation in SCLISP 2 corrects a number of other problems that existed in previous Lisp releases. To simplify the process of controlling memory usage within Lisp, the installation procedure now includes settings for the amount of dynamic and reserved space that will be left free in the Lisp image after a DISKSAVE. Making the dynamic space bigger increases virtual memory usage, but it reduces the frequency of garbage collection. Thus if the customer site has sufficient disk, there are more easy to use options for tuning performance. Now there is also provision for installing site local customization. If a site has a library of code it will always want to use, it can be added to the Lisp image at the time Lisp is installed. Sun uses this feature to install patches to the system.

Configuring Lisp

Also with simplicity of use in mind, the process of configuring the Lisp system for installation can now be done with a SunView program included on the distribution tape. This program creates a window in which all the information necessary to configure Lisp is entered into panels and editor subwindows. In addition, each section in the configurator has on-line help. The file produced by the configurator is read in by Lisp during installation.

Window Tool Kit

Of the new components in SCLISP, perhaps the most significant is the Window Tool Kit. This facility gives the Common Lisp programmer the capability to create a variety of interactive interfaces in much the same way that it is done on dedicated Lisp Machines. The tool kit is compatible with SunView in that the region it occupies on the screen is a SunView window, but inside what Lisp considers its screen, the window manager is much different, and in some key areas more powerful than SunView.

The Lisp Window Tool Kit allows the programmer to deal with the screen at several levels of abstraction. It can be looked upon as a simple viewport to the underlying frame buffer. A complete set of rasterops are provided for managing bitmaps. This level of access gives the programmer the ultimate flexibility in controlling how the screen looks. Above the viewport level, there is the notion of a window hierarchy. Windows with borders and titles can be created. It is also possible for a window to have embedded child windows. I/O to windows is done via an extension to the STREAM mechanism in Common Lisp. Thus the standard FORMAT command may be used to display characters. Bit level operations may be done by opening a bitmap stream.

One major departure in the Lisp Window Tool Kit from SunView is that the Window Manager is recursively re-enterable. New windows can be created at any time, and Lisp Windows can be used to debug a program that in turn is using the Window Tool Kit. A second important difference is that Lisp Windows can have arbitrary active regions. Specialized processing can be defined whenever the mouse enters or leaves a region. In addition there is provision for the more normal button events. Finally, the Lisp Listener stays active even when the Window Tool Kit is being used. So the Lisp interpreter can be used to display new windows or try out a new interaction sequence.

Lisp Editor

The Editor that comes with SCLISP 2 is a Lisp-based version of Emacs. It is similar to the editor that is part of the Symbolics Lisp Machine software. In addition to the usual set of charater-oriented editing features, the Lisp Editor has a number of specialized features for Lisp. These include things like parenthesis balancing, pretty-printing and a set of commands for quickly moving around s-expressions. In addition there are commands for compiling definitions and evaluating forms that can be executed directly from an editing buffer.

The Lisp Editor will work either on a standard terminal like a Shell Window or a Wyse Terminal, or it can be used under the Lisp Window Tool Kit. In either case, when the editor is started, one of its buffers is hooked up to the Lisp Listener. This provides a way for the user to communicate with Lisp's top level through the Editor. Finally, the Editor is programmable. New commands can be created by writing new functions in Common Lisp and hooking these functions to Editor recognized key bindings. Executing a user written command is the same as executing one delivered with the system.

Flavors

Flavors is an object-oriented extension to Lisp that has been in use on Lisp Machines for several years. It gives the programmer a way to structure a complex application in terms of abstract objects with local state and methods. The program executes by having objects send each other messages. This style of programming is becoming very popular because of the way it tends to encapsulate complex operations. The interface between modules becomes a message protocol rather than remaining dependent on a specific set of procedural and data type relationships.

The implementation of Flavors included with SCLISP 2 is quite close to the software developed at MIT and included with Symbolics system releases up through RELEASE 6. The current Symbolics software, RELEASE 7, has a new object system called New Flavors. It is a competitor to Xerox's Common Loops and is quite different from the Flavors in our Lisp release. It is unclear what impact New Flavors will have on the Common Lisp community beyond influencing the design of stand object system for Common Lisp, which at this point looks like it will be based on Common Loops. Common Loops is not supplied on the SCLISP2 distribution tape, but a version of Common Loops that runs under SCLISP 2 is available over the ARPA net.

Foreign Call Mechanism

In addition to the new features, there are some important enhancements to existing features and performance in SCLISP 2. The Foreign Call mechanism has been reworked and is now much cleaner. Rather than relying on the use of global variables to hook up foreign code to Lisp, it is now possible to explicitly define a Lisp function that is implemented as C or FORTRAN code. For compatibility, we have left the old mechanism in for the time being, but it will be pulled out once customers have had a chance to convert their programs.

Performance, Ease of Use, etc.

The Lisp Compiler is quite a bit faster in this release than in the previous version of SCLISP, particularly in smaller memory machines. This was accomplished by tuning the locality of the Compiler's code so that paging is minimized. It is now possible to make use of the Compiler on a small machine like a 3/50. We don't recommend trying to use the full Lisp Development environment on less than an 8 megabyte machine, however.

To cut down dependence on the version of Sun Unix used with our Lisp product, the libraries needed to support the Window Tool Kit, Editor and (soon) Floating Point are now bound at the time the Lisp system is installed at the user's site. This makes the installation process a little slower, and it now takes more disk than it used to. But on the positive side, it sets things up so that this Lisp release will run on systems with new frame buffers and special hardware like FPA cards.

The last really obvious new feature in this release is the documentation. A new expanded User's Guide replaces the old one. It describes all the features in the system that extend the Common Lisp standard. Included in the User's Guide are descriptions of all the debugging tools, Windows, Flavors, Editor etc. This new manual is also rich with examples. That should help people get started. The Steele book describing the Common Lisp standard has been replaced with a new Common Lisp Reference Manual. This book clarifies many of the ambiguitics present in the Steele report. It is also layed out in a manner which makes it much easier to find things. Like the new User's Guide, the Reference Manual contains a number of examples. A guide to installing Common Lisp and an RTF (Read This First) round out the documentation package.

Besides the officially suported Lisp product, there are some unsupported extras. We wanted our users to have an easier time getting started with Common Lisp. Toward this end we have included a directory of 'goodies' on the Lisp distribution tape. In it are some examples, with documented source, on how to use some of the new features in the Window Tool Kit and Flavors packages. There is also an interface to SunView that allows Lisp programs to use SunView in the same ways a C program might. In future releases we intend to further enhance this directory with more useful demos and examples. Finally, there is an enhanced version of our Unipress Emacs (Emacs 2.10) interface for those customers who would rather not use the built-in Lisp Editor.[2]

[2] The above article was submitted by Steve Gadol, Manager of Symbolic Computing, whose last word on the topic is:

```
(happy (lisp (hacking)))
```

Revision History

Revision	***Date***	***Comments***
FINAL 5	March 1987	Second issue of Software Technical Bulletin (Software Information Services).