



# 66

# AIX

*April 2001*

---

## **In this issue**

- 3 What do you mean, “File Not Found”?
- 7 32-bit versus 64-bit architectures
- 12 AIX Performance monitoring using the GlancePlus Motif interface and Adviser features
- 23 Smit tutorial script
- 35 Standby system failover
- 52 AIX news

---

© Xephon plc 2001

update

# AIX Update

---

## Published by

Xephon  
27-35 London Road  
Newbury  
Berkshire RG14 1JL  
England  
Telephone: 01635 38342  
From USA: 01144 1635 38342  
E-mail: [trevore@xephon.com](mailto:trevore@xephon.com)

## North American office

Xephon  
PO Box 350100  
Westminster, CO 80035-0100  
USA  
Telephone: 303 410 9344

## Subscriptions and back-issues

A year's subscription to *AIX Update*, comprising twelve monthly issues, costs £180.00 in the UK; \$275.00 in the USA and Canada; £186.00 in Europe; £192.00 in Australasia and Japan; and £190.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the November 1995 issue, are available separately to subscribers for £16.00 (\$23.00) each including postage.

## AIX Update on-line

Code from *AIX Update* can be downloaded from our Web site at <http://www.xephon.com/aixupdate.html>; you will need to supply a word from the printed issue.

## Editors

Trevor Eddolls and Richard Watson

## Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, scripts, and other contents of this journal before making any use of it.

## Contributions

When Xephon is given copyright, articles published in *AIX Update* are paid for at the rate of £170 (\$260) per 1000 words and £100 (\$160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 (\$80) per 100 lines. In addition, there is a flat fee of £30 (\$50) per article. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*, or you can download a copy from [www.xephon.com/contnote.html](http://www.xephon.com/contnote.html).

---

© Xephon plc 2001. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*

## What do you mean, “File Not Found”?

One of the delights of being a Unix systems administrator, with servers running third-party software, is having to deal with the infamous error message “Error number #2: file not found”. (The “#2” comes from the operating system, where 2 is the ENOENT error, as in */usr/include/errno.h*). And of course, as with most third-party software, there will probably be no error log informing you which file is missing. A call to the company’s Help Desk will probably get you a reply similar to “Our software is bug-free so you must have deleted the file yourself” – yeah, right!

So, faced with an unhelpful third-party Help Desk and an angry user/set of users, where’s best to turn? One of the best options I have found is to use the AIX system trace, providing of course that the user can, and is willing to, reproduce the error. Note however that using the system trace on a busy system can generate a trace file of many megabytes of data to sift through. So, having chosen this approach (usually because there is no other option left) how best to proceed?

Tracing is best turned on using *smit*. This gives you the advantage of a pick-list for the selection of AIX system actions to trace. For “File Not Found” problems, I suggest choosing *fact* and *filephys* options. The other *smit* options I usually alter are:

- The output filename (to point to a directory with a large amount of space).
- To **stop** when the trace file is full, otherwise the part of the trace you are most interested in could be over-written (if you don’t change this option, check for the string *wrapped* in the trace report).
- The **size** of both the **trace buffer** and the **trace file**.

Before starting the trace, it is worth getting the user to the point where one mouse click/key press will generate the error. Now start the trace, ask the user to generate the error, and stop the trace as quickly as possible after the error message (either via *smit* or using the *trcstop* command).

To translate the binary trace output into readable form, either use *smit* again or the *trcrpt* command.

Below are two examples of trace output (both have been slightly edited). The first shows the running of the *xhost* command. It shows the *ksh* process searching for the *xhost* program, finding it in */usr/bin/X11*, and running it (*exec: cmd=xhost...*).

```
Thu Jan  4 06:35:30 2001
System: AIX test Node: 4
Machine: 002146058900
Internet Address: 80000001 128.0.0.1
The system contains 1 cpus, of which 1 were traced.
Buffering: Kernel Heap
This is from a 32-bit kernel.
```

```
/usr/sbin/trace -j
106,134,139,107,135,15b,12e,19c,163,106,134,139,107,135,15b,12e,19c,163,10a
-o /usr/pp/trc.out -a
```

ID	PROCESS	PID	ELAPSED	SYSCALL	KERNEL	INTERRUPT
001	--1-	4304	0.000000		TRACE ON channel 0	Thu Jan  4 06:35:31 2001
106	ksh	7926	4.861316		dispatch: cmd=ksh pid=7926	
				tid=9219 priority=60 old_tid=517 old_priority=127 CPUID=0 [5216 usec]		
19C	ksh	7926	4.861591	write(2,2004D3E8,3)		
163	ksh	7926	4.862193	read(0,2FF21854,1)		
106	ksh	7926	5.199458		dispatch: cmd=ksh pid=7926	
				tid=9219 priority=60 old_tid=517 old_priority=127 CPUID=0 [129931 usec]		
19C	ksh	7926	5.200057	write(2,2004D3E8,1)		
163	ksh	7926	5.200684	read(63,2FF22620,400)		
10A	ksh	7926	5.200699		PFS rdwr	
				(vp,ip)=(1323D600,1323D6300)		
19C	ksh	7926	5.200958	write(63,20052748,8)		
10A	ksh	7926	5.200965		PFS rdwr	
				(vp,ip)=(1323D600,1323D630)		
10A	ksh	7926	5.200977		PFS writei VA.S=0000 1490.2669	
				bcount=0008 ip=1323D630		
107	ksh	7926	5.200977*		lookupn: /usr/bin/xhost	
107	ksh	7926	5.201508		lookupn: file not found	
107	ksh	7926	5.201508*		lookupn: /etc/xhost	
107	ksh	7926	5.201665		lookupn: file not found	
107	ksh	7926	5.201665*		lookupn: /usr/sbin/xhost	
107	ksh	7926	5.201859		lookupn: file not found	
107	ksh	7926	5.201859*		lookupn: /usr/ucb/xhost	
107	ksh	7926	5.202052		lookupn: file not found	
107	ksh	7926	5.202052*		lookupn: /usr/bin/X11/xhost	
139	ksh	7926	5.202435*	fork:	pid=11208 tid=10595	

```

106 ksh      11208  5.204434      dispatch:  cmd=ksh pid=11208
tid=10595 priority=60 old_tid=9219 old_priority=60 CPUID=0 [4975 usec]
106 ksh      7926   5.206267      dispatch:  cmd=ksh pid=7926
tid=9219 priority=60 old_tid=10595 old_priority=60 CPUID=0 [1833 usec]
106 ksh      11208  5.208302      dispatch:  cmd=ksh pid=11208
tid=10595 priority=60 old_tid=9219 old_priority=60 CPUID=0 [2034 usec]
12E ksh      11208  5.209073      close fd=10
107 ksh      11208  5.210340*     lookuppn: /usr/lib/nls/msg/en_US/
ksh.cat
15B ksh      11208  5.210850     open /usr/lib/nls/msg/en_US/ksh.cat fd=3
RONLY
163 ksh      11208  5.210987     read(3,20008A60,1000) /usr/lib/nls/msg/
en_US/ksh.cat
12E ksh      11208  5.211587     close /usr/lib/nls/msg/en_US/ksh.cat fd=3
107 ksh      11208  5.211587*     lookuppn: /usr/bin/X11/xhost
10A ksh      11208  5.215350     PFS rdwr
(vp,ip)=(132689A0,132689D0) /usr/bin/X11/xhost
10A ksh      11208  5.215361     PFS readi  VA.S=0000 0000.4F73
bcount=0100 ip=132689D0 /usr/bin/X11/xhost
134 ksh      11208  5.215438     exec:  cmd=xhost pid=11208 tid=10595
106 xhost    11208  5.216365     dispatch:  cmd=xhost pid=11208
tid=10595 priority=61 old_tid=10595 old_priority=61 CPUID=0 [8063 usec]

```

The second extract shows the system searching for the command *tester*. Because this command does not exist, it never produces the *exec* or *fork* for the program. Notice that you can determine the search path for the program, hence you can work out the \$PATH environment variable for the user entering the command. This can be very useful.

```

107 ksh      10086   8.433221*    lookuppn: /usr/bin/tester
107 ksh      10086   8.434067     lookuppn: file not found
107 ksh      10086   8.434067*    lookuppn: /etc/tester
107 ksh      10086   8.441534     lookuppn: file not found
107 ksh      10086   8.441534*    lookuppn: /usr/sbin/tester
107 ksh      10086   8.442104     lookuppn: file not found
107 ksh      10086   8.442104*    lookuppn: /usr/ucb/tester
107 ksh      10086   8.442324     lookuppn: file not found
107 ksh      10086   8.442324*    lookuppn: /usr/bin/X11/tester
107 ksh      10086   8.442745     lookuppn: file not found
107 ksh      10086   8.442745*    lookuppn: /sbin/tester
107 ksh      10086   8.442935     lookuppn: file not found
107 ksh      10548   8.447553*    lookuppn: /usr/lib/nls/msg/en_GB/ksh.cat
107 ksh      10548   8.448593     lookuppn: file not found
107 ksh      10548   8.448593*    lookuppn: /usr/lib/nls/msg/en_US/ksh.cat
107 ksh      10548   8.448816*    lookuppn: /usr/lib/nls/msg/en_US/ksh.cat
107 ksh      10548   8.452487*    lookuppn: /usr/bin/tester
107 ksh      10548   8.455533     lookuppn: file not found
107 ksh      10548   8.457033*    lookuppn: /etc/tester
107 ksh      10548   8.457405     lookuppn: file not found

```

```
107 ksh      10548      8.457605* lookuppn: /usr/sbin/tester
107 ksh      10548      8.459069  lookuppn: file not found
107 ksh      10548      8.464338* lookuppn: /usr/ucb/tester
107 ksh      10548      8.464489  lookuppn: file not found
107 ksh      10548      8.464489* lookuppn: /usr/bin/X11/tester
107 ksh      10548      8.466056  lookuppn: file not found
107 ksh      10548      8.466056* lookuppn: /sbin/tester
107 ksh      10548      8.467363  lookuppn: file not found
107 ksh      10548      8.467538* lookuppn: /tmp
107 ksh      10548      8.468169* lookuppn: /tmp
```

The quickest way to sort through a large trace report is to *grep* for the string *lookuppn* (as above). This way you can find out which programs are being run and can track down the *PID* you are interested in.

I have used this method of debugging problems too many times to count. It has helped track down problems in IBM code as well as third-party software and even my own programs. To start with, on a quiet system, run a couple of commands whilst trace is active and have a look at the output. You may even be able to work out how some of the AIX programs are coded.

I hope it proves as useful to you as it has to me.

---

*Phil Pollard*  
*Unix Systems Administrator (UK)*

© Xephon 2001

---

## **Code from *AIX Update* articles**

Code from individual articles of *AIX Update* can be accessed on our Web site, at:

<http://www.xephon.com/aixupdate.html>

You will be asked to enter a word from the printed issue.

## 32-bit versus 64-bit architectures

If you upgrade the machine your application runs on, from a 32-bit machine to a 64-bit machine, your users will revel in the new-found power and performance of their application, right?

*Not!* Well, maybe. There are some common misconceptions about 64-bit computing and what it does and does not offer. There are three main issues to consider when defining a 64-bit environment. Is the hardware 64-bit? Will my operating system support and exploit 64-bit hardware? Are my applications poised to support and/or exploit a 64-bit environment?

### 64-BIT HARDWARE DEFINED

64-bit CPUs manipulate data in 64-bit chunks, ie the instruction set of the computer handles 8-bytes of data by design. This is distinct from some 32-bit machines where they *handle* 64-bit long data, and have instructions for doing so, but under the covers it is done in 32-bit chunks.

64-bit CPUs use 64-bit *physical* memory addresses to address memory space. Obviously this provides the ability to have large amounts of directly accessible RAM.

These are the two distinguishing features of 64-bit hardware. Back to the question in my opening paragraph then. If you ‘push-pull’ a machine of equal characteristics (RAM, CPU speed, etc) from 32-bit to 64-bit, will you achieve an immediate performance boost? Probably not. The assumption that this will occur stems from the fact that you can address more real memory on 64-bit architectures, and, of course, more real memory allocated to database applications and the like almost always means a measurable performance improvement. If you are not adding memory, the chances are that no measurable difference will be seen. There are some qualifications on this, discussed later.

## IS MY MACHINE 64-BIT?

The following is a list of the most common 64-bit RS/6000 machines:

- SP/2 375 MHZ Thin, Wide and High Nodes – Power3-II.
- pSeries 640 and 680.
- Models 7026-M80, H80, H70.
- Models S70, S7A, S80.
- Model 7025-F80.
- Model 7043-260.
- Model 7044-170,270.

Another ‘clue’ to help determine if your machine is 64-bit, if it is not on the above list, is to:

```
ls1pp -l bos.64bit
```

This fileset is automatically installed by AIX on 64-bit machines. One other clue might be checking */etc/inittab* for ‘load64bit’. This is the entry created when you run */etc/methods/cfg64* or smitty ‘Enable 64-bit Application Environment’.

There is nothing special to consider when installing AIX 4.3.3 or 5L on a 64-bit machine.

AIX is ‘smart’ enough to detect the 64-bit architecture and install the appropriate 64-bit kernel programs.

## AIX OPERATING SYSTEM SUPPORT – FULL BINARY COMPATIBILITY

There is only *one* version of AIX for both 32-bit and 64-bit machines.

AIX 4.3.3 and AIX 5L both provide complete side-by-side compatibility for 32-bit and 64-bit applications. With very few exceptions, a 32-bit program will run without changes in a 64-bit environment. Any documentation from IBM on this subject that I have read reinforces their commitment to making 32-bit to 64-bit transitions as seamless as possible. The few exceptions mentioned usually involve 32-bit



device drivers interoperating in a 64-bit environment.

The details of these exceptions are outside the scope of this article. More reading can be done on them in the references provided at the end of this article.

Note: obviously a program compiled in 64-bit mode *will not* run on a 32-bit machine.

## APPLICATION ISSUES

Applications that manage 2GB of address space or less do not need to be recompiled or re-engineered for a 64-bit environment. There will be little or no benefit in doing so. In fact, it's possible to see *minor* performance hits by just taking a program and recompiling it in 64-bit mode.

This is because of a larger program executable size and hence slightly longer loader times.

To give a simple example of this, consider the basic Hello World! program:

```
#include <stdio.h>
main ()
{
    printf("hello world!\n");
}
```

If we compile this in both 32-bit mode and 64-bit mode the resulting binary will be approximately 10% larger for the 64-bit program:

```
->cc hello.c -o hello32 -q32
->cc hello.c -o hello64 -q64
->ls -al hello*
-rwxr-xr-x  1 root      system      4083 Jan 29 18:47 hello32
-rwxr-xr-x  1 root      system      4492 Jan 29 18:48 hello64
```

While this seems relatively insignificant for our Hello program, applied to an environment with a large number of 'popular' binaries and frequent program loads of the same, it could make a measurable difference.

There are numerous optimization strategies and considerations for migrating and compiling 32-bit programs for 64-bit environments.

The chapter on this in the redbook *AIX 64-Bit Performance in Focus* is excellent.

## ORACLE AND DB2

Oracle 8.1.x will ship for AIX with install media for either 32-bit or 64-bit. The 32-bit version will run just fine on a 64-bit machine. The 64-bit version is recompiled in 64-bit mode and should be considered only if you have a requirement to address large memory spaces.

Later releases of DB2, from what I have read, are already poised to take advantage of large memory spaces in a 64-bit environment 'out of the box'.

## 64-BIT STANDARDS

There are two sets of relevant standards for 64-bit computing in the Unix world. The first is the Single Unix Standard, Version 2. Second is the industry agreement on managing large data sizes, referred to as LP64. There is great detail on these standards at the freebsd Web site (<http://www.unix-systems.org/version2>). The Common Hardware Reference Platform (CHRP), which is a specification related to the PowerPC architecture, also has specifications for 64-bit.

## LARGE FILE SUPPORT

Even though large file (>2GB) support has been provided since AIX 4.2 (ie with 32-bit programs), the ability to address files >2GB requires addresses larger than 32 bits. A true 64-bit environment (hardware and O/S) provides this capability much more efficiently.

## SUMMARY

Given the smooth transition that AIX provides and the fact that most 64-bit RS/6000s are priced competitively (new) with comparable 32-bit systems, it is a worthwhile exercise at this stage to plan new machine purchases and upgrades with 64-bit computing in mind. You may not need the larger memory spaces and other benefits today, but will be prepared down the road.

## REFERENCES

- *AIX 64-Bit Performance in Focus*, IBM Redbook number SG24-5103 (<http://www.redbooks.ibm.com/redbooks/SG245103.html>).
- *RS/6000 Systems Handbook – 2000 Edition*, IBM Redbook number SG24-5120 (<http://www.redbooks.ibm.com/redbooks/SG245120.html>).
- <http://www-1.ibm.com/servers/eserver/pseries/hardware/enterprise/>.
- <http://www.rs6000.ibm.com/resource/technology/chrp/index.html>.
- *The RS/6000 64-bit Solution* IBM RS/6000 Web site <http://www.rs6000.ibm.com/resource/technology/64bit6.html>.

---

David Miller  
Database Architect  
Baystate Health Systems (USA)

© Xephon 2001

### **Need help with an AIX problem or project?**

Maybe we can help:

- If it's on a topic of interest to other subscribers, we'll commission an article on the subject, which we'll publish in *AIX Update*, and which we'll pay for – it won't cost you anything.
- If it's a more specialized, or more complex, problem, you can advertise your requirements (including one-off projects, freelance contracts, permanent jobs, etc) to the hundreds of AIX professionals who visit *AIX Update's* home page every month. This service is also free of charge.

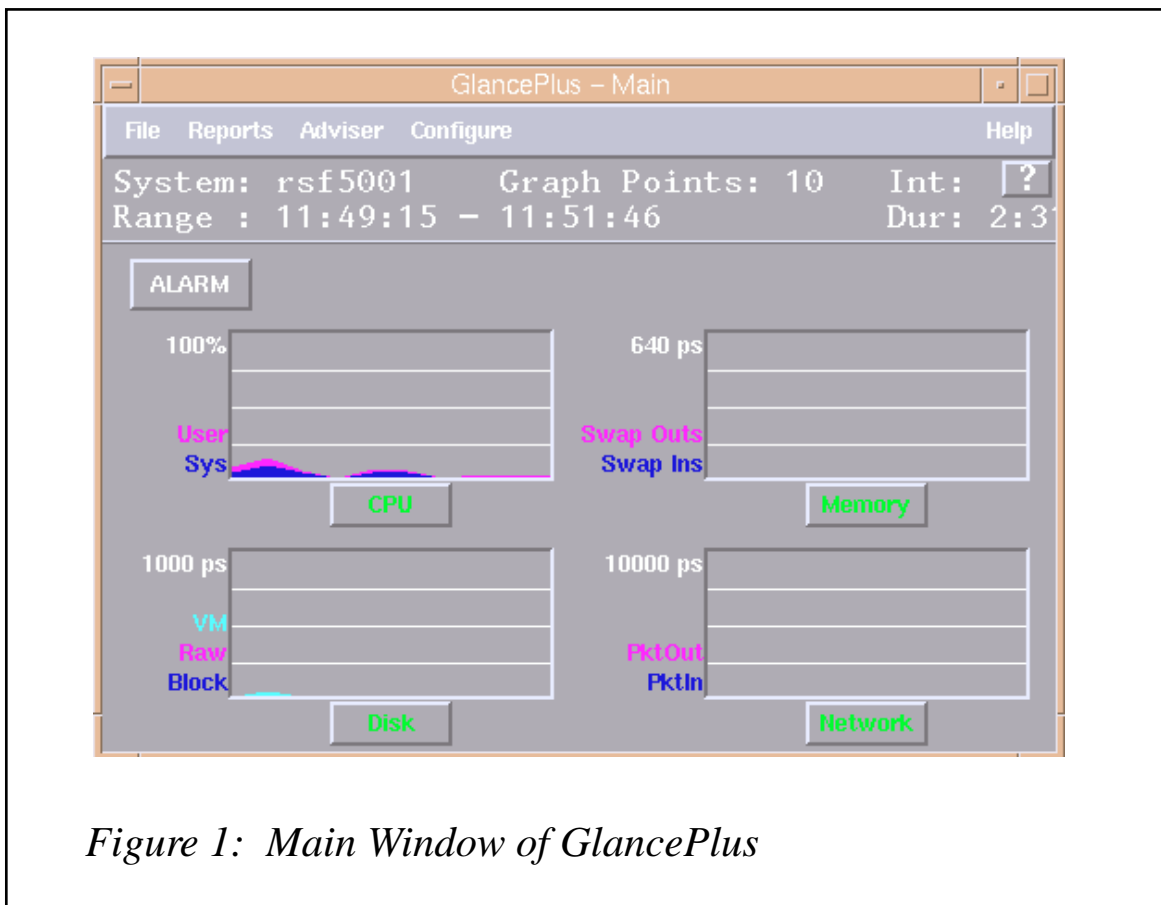
Visit the *AIX Update* Web site, <http://www.xephon.com/aixupdate.html>, and follow the link to *Suggest a topic* or *Opportunities for AIX specialists*.

# AIX Performance monitoring using the GlancePlus Motif interface and Adviser features

GlancePlus is a system performance monitoring and alarm tool produced by Hewlett-Packard. My previous article (*Performance monitoring with GlancePlus, AIX Update, Issue 62, December 2000*) described a curses-based interface to this product. This article will cover two advanced features in the product – Motif-based Graphical MMI, invoked by the command **gpm**, and the Adviser module, which enables the monitoring of system conditions according to user-defined rules.

## GPM REPORTS

When the Motif interface of GlancePlus, **gpm**, is started, a graphical window is displayed containing four windows, which display graphs



*Figure 1: Main Window of GlancePlus*

that represent the four most important performance aspects of the system – CPU, memory, disk, and network. See Figure 1.

A button, located under each of the graphs, shows the state of the alarm associated with each window. The colour of the button indicates the state of the alarm. When everything's OK it's green, for warning it's yellow, and when critical it's red.

The selection of each of the buttons enables a display of the graphs containing different performance data, related to the button's window. The selection of the **ALARM** button, located in the upper-left part of the display, invokes the display of a window containing history details of past alarms.

Two lines of the display, located under the menu bar of the **gpm** window, contain general information such as name of the monitored system, number of displayed graph points, length of the measurement interval, range of time displayed in the graph windows, and total duration of **gpm** invocation. The icon marked by a question mark, which is located in the upper-right corner of the main window, enables the invocation of a hypertext-based help system. First you select this icon, then point with your pointer on the part of the display that interests you and press the left mouse button. The window containing the requested help appears on your display.

The *Main* menu buttons of **gpm** enables the selection of the following features:

- *File Menu* is used to perform operations that influence all the displayed windows.
- *Update Now* enables the immediate measurement and display of performance statistics.
- *Reset Cum To Zero* resets the cumulative statistics to zero.
- *Iconify All* places all open windows into a single icon. When you click on the icon, the windows are redisplayed.
- *Conceal Window* is used to remove the window but leave all its dependent windows visible.
- *Exit GlancePlus* causes the immediate exit from the **gpm**.

*Reports* Menu is used to select and display various **gpm** reports:

- *Resource History* enables the display of a window containing CPU, disk, network, and memory graphs containing data points collected during the last ‘*n*’ samples. The look of this window can be changed with *Configure*. Each graph has several lines, representing each resource, eg the CPU graph shows user and system CPU utilization.
- *CPU Info* points to a submenu enabling the display of the CPU graph, CPU report, and CPU-by-processor. The CPU graph depicts the CPU queue length over time. The CPU report is a table of various global CPU metrics, such as types of CPU utilization, the CPU queue length, and the system call rate. This report is identical to the **glance CPU (c)** screen. The CPU report is most useful in a Uniprocessor environment. The CPU-by-processor report is most useful in a multiprocessor environment – it shows CPU utilization, the one-, five-, and fifteen-minute load averages, the context switch rate, fork rate, and the last PID, all by processor number. It is the same as the **glance All CPU (a)** screen.
- *Memory Info* points to a submenu enabling the display of the memory graph, memory report, and virtual memory graph. The memory graph shows page ins, page outs, and swap outs per second. The memory report displays tabular data showing the breakdown of memory activity and the type of memory available for various OS needs. The metrics are shown as current, cumulative since **gpm** started, values, and percentages. This report is similar to the **glance Memory (m)** screen.
- *Disk Info* points to a submenu enabling the display of the disk graph, disk report, I/O-by-disk, file system capacity, and logical volume info. The disk graph shows the top ten disks by utilization. The disk report shows global information on the types and rates of disk activity and is similar to **glance Disk (d)** screen. The I/O-by-disk report lists the following metrics: disk queue length, physical, logical, and virtual memory, file system, and system, and raw I/O rates. This screen is similar to the **glance Disk Queue (u)** screen. The file system capacity screen displays information

about the utilization of the file systems. The logical volume info report displays information describing the logical volume definition metrics.

- *Network Info* points to a submenu enabling the display of the network graph, network-by-card graph, network-by-interface, NFS global activity, and NFS-by-operation. The network graph window shows network packet I/O rates and percentage network errors. The network-by-interface window displays input and output packets, and error rates for each of the available network interfaces. The information displayed by this screen is similar to the **glance** *Network (l)* screen. NFS global activity shows Network File System activity from *this system's* perspective, as well as metrics such as read and write queues, idle BIODs, and response times. It is similar to the **glance** *global NFS (N)* screen. The NFS-by-operation report shows the various access methods used by NFS in, for example, remote file access and remote procedure calls.
- *System Info* points to a submenu enabling the display of the system tables graph, system tables report, and system attributes. The system tables graph lets you see at a glance the utilization, both current and maximum, for each of your system tables. The system tables report provides the utilization information in tabular format, and is similar to the **glance** *System Tables (t)* screen.
- *Swap Space* enables the display of a window containing swap space configuration and utilization data. Its is similar to the **glance** *Swap (w)* screen.
- *Wait Queue Graphs* enables the display of a window containing four graphs: CPU/Q-length, SWAPS/Q-length, VM wait queues, and I/O queue. An example is shown in Figure 2.
- *Transaction Tracker* displays a tabular report of the transactions reported by applications built with the HP-supplied Transaction reporting API.
- *Application List* reports data describing groups of processes that do similar work. Applications are defined in a file external to

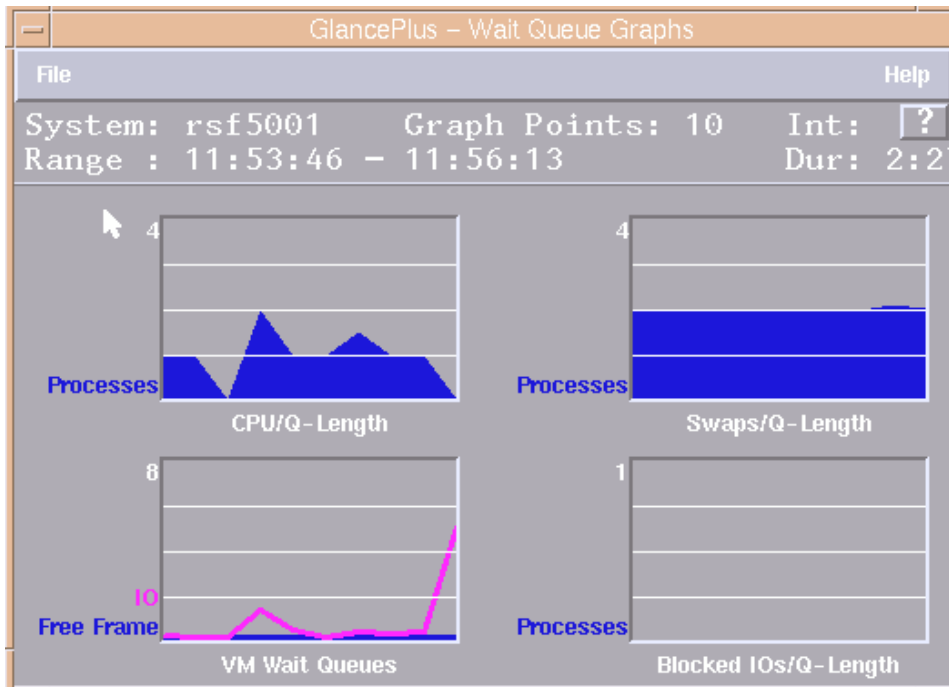


Figure 2: Wait queues graph window of GlancePlus

GlancePlus called the 'parm' file. The file and its format are described in detail in the *Defining Application* topic of **gpm**'s help. The *Applications List* window shows the applications defined in the 'parm' file and their corresponding metrics. Double-clicking on an application's name shown in the *Application List* will invoke a display of the window containing the processes that make up an application. Use the *Reports* menu on the *Applications List* window to view the Application CPU graphs window. It displays a graph for each application's CPU utilization.

- *Process List* displays a window containing a table of processes currently running on the system. This table contains process names, user names, and associated metrics that define the load each process is putting on the system. This screen is similar to the *Global (g)* screen of **glance**. When you highlight a process from the list, you have access to the options under the Reports and Admin menus. The Report menu enables the display of detailed information about the selected process. The Process Resources



Process Name	PID	CPU %	IO Byte Rate	Block IO Rate	State Reason	Prio
smssed	10342	0.1	0.0	0.0	SLEEP	0
dswm	73680	1.0	3.1	0.0	SLEEP	0
dlserv	64796	0.5	0.7	0.0	SLEEP	0
orac1eGTENCO	96026	0.5	0.0	0.0	SLEEP	0
http	89100	0.3	5.2	0.0	new	0
ppm	79264	0.1	11.9	0.0	SLEEP	0
Oracle	37766	0.1	0.0	0.0	SLEEP	0
ora_ckpt_001	82058	0.1	0.0	0.0	SLEEP	0
ora_dbw0_001	80790	0.0	0.0	0.0	SLEEP	0
ora_p001_001	81302	0.0	0.0	0.0	SLEEP	0
ora_ckpt_001	82058	0.0	0.0	0.0	SLEEP	0
ora_mmon_001	83718	0.0	0.0	0.0	SLEEP	0

Figure 3: Process List window of GlancePlus

window shows CPU, disk, memory, and other specific measurements concerning the process. This screen is similar to the **glance** *Single Process (s)* screen. The *Process Open Files* window shows files opened by the process along with the mode in which the file was opened. This screen is similar to the **glance** *Open Files (F)* screen. The *Process List* screen is shown in Figure 3.

If, after reviewing the process, you need to either ‘kill’ it or assign a new value to it, you can select the appropriate option from the *Admin* menu.

## MONITORING YOUR SYSTEM WITH THE ADVISER

The GlancePlus Adviser provides a way to automate monitoring and responding to the state of key performance indicators. At each measurement interval, the Adviser executes a predefined set of instructions that examine performance metrics. The Adviser bases its decisions on user-specified symptoms that are defined in terms of the interval time periods and alarms that are defined for longer periods of

time. Symptoms and alarms can ease performance management of complex systems by warning operators of new performance problems by sending alerts. The Adviser can be assessed and configured through the following menu items:

- *Symptom Status* shows the status of all symptoms during the current interval. This window is updated during each interval.
- *Symptom Snapshot* shows the status of all symptoms during the current interval. This window is not updated during each interval.
- *Symptom History* displays a history of each symptom defined in the Adviser Syntax. This lets you see if the Adviser has determined that there are continued problems for a resource over a period of time. There is one graph for every SYMPTOM statement, and, within each graph, one bar for each interval in which the symptom was evaluated.
- *Alarm History* displays a log of events that have occurred since the start of the **gpm** run. This window displays up to 250 alert events.
- *Edit Adviser Syntax* causes the display of a separate window containing a text editor that enables the editing of the actual syntax of the alarms. If you prefer, the syntax can be saved to a file, edited with the editor of choice, and then reloaded into **gpm**. Alarm syntax consists of a symptom name, a probability value, a time period, an alert for a major resource, and a reset value.

The following is an example of how to code an alarm based on the size of the run queue:

```
if runqueue > 20 for 5 minutes {
    start YELLOW ALERT CPU
    reset
}
```

You can also edit rule-based symptoms, by selecting the *Symptoms Window* display from the *Adviser Syntax* window menu. Symptoms are named, and consist of a collection of rules that are based on specific metrics. The exact metric names can be found using the on-line help facility, or you can create your own variable metrics. For

example, to create a rule based on the size of the CPU run queue, one would define a new symptom variable called **runqueue** and set it to a probability of 75 if the length of the CPU run queue exceeds 10:

```
SYMptom runqueue
RULE gbl_pri_queue > 10 probability 75
```

Additional menu entries in the *Adviser Syntax* window enable you to check and install the syntax edited by you and reset the syntax to its default state.

## CONFIGURING GLANCEPLUS

The graphs and reports of **gpm** can be configured in much more advanced ways than similar reports in **glance**. Most of the windows that display tabular information reports have a *Configure* menu containing the following options:

- *Sort Fields* displays a window that will enable you to select report columns by which the displayed data is going to be sorted. The selection is done in an interactive context-help guided fashion. For instance, you can change the default order of the display of *Processes* to be sorted by the number of I/O operations instead of by CPU time.
- *Arrange Columns* displays a window that enables you to arrange the order of report columns in which the data is going to be displayed. The selection is done in an interactive context-help guided fashion. For instance, you can change the default order of the display of *Processes* to display process priority instead of CPU consumption as the second column of the table.
- *Filters* displays a window that enables you to specify criteria for values of information fields that will be displayed in the report. You choose metrics by which you want to filter the report, then the relation that should be used, and then the value that is the base of the relation. For instance, you can choose a display of processes that belong to a certain user and consume more than 5% of CPU time.
- *Choose metrics* displays a window that enables you to select

specific metrics that will appear in the report window. This is a very significant feature enabling you to totally change the data that appears in the report according to your needs. For instance, you can display only I/O-related statistics of the *Processes*, if you are trying to find the most active I/O users of your system.

The *Configure* menu of **gpm** contains the following options that modify the global display characteristics of the tool. This menu contains the following options:

- *Colors* causes the display of a window that enables you to change the colours used for the graphs and process filters. If desired, you can also change the *Adviser* colours. Depending on the availability of colours, **gpm** may not be able to reserve its own private colours. If this is the case, and **gpm** uses shared colours, you will not have access to the *Colors* option.
- *Font* causes the display of a window that enables you to change the size of the font used to display the report windows. Decreasing the font size allows you to see more columns of the report at one time. Changing the font size does not affect the font used to label graphics.
- *Measurement* causes the display of a window that enables you to define the length of the measurement interval and the number of these intervals to save in the history buffers. The interval defines the time between measurement updates. The history buffers save the pertinent interval information for use in graphs and the *Application History* window. For graphs that show history, the history buffer dictates the number of points shown on the graph, with each point representing one interval.
- *Icon* causes the display of a window that enables you to define the type of the graph that depicts CPU statistics and is displayed when the **gpm** is iconified, inside the icon. You have the following options: system history, CPU states current, CPU states history, system states stacked, and system Kiviat graph.
- *Graph Limits* (for *Main* graph) – the disk and network graphs scale automatically when a data point is measured that is higher than the current high value (which is the highest value that **gpm**

has observed since it was installed). If the top point of these graphs is an unusual spike, you may want to reset the limit to a lower value. The limit you enter must always be equal to or higher than the highest measured point found in the history buffer. If the point in the history buffer is higher than the input value, the highest measured point is used as the limit value.

- *Main Graph* opens a submenu that enables you to change the style of the *Main Graph* of **gpm** to one of following values: horizontal bars, vertical bars, pie charts, and resource history.

## COMPARISON OF GPM AND GLANCE

The **gpm** interface of GlancePlus has some significant advantages when compared to **glance**. These include:

- Graphical presentations of data.
- Extensive usage of colours to represent severity of alerts.
- Extensive tabular data filtering and sorting.
- Ability to display/hide selective data items in tabular reports.
- Customizable alarms and symptoms.

On the other hand, **gpm** has some disadvantages such as:

- Significant memory consumption
- Very limited logging capability.

As with **glance**, the CPU overhead incurred by **gpm** depends on the refresh interval and the number of running processes. Longer intervals cause less overhead. Additional CPU overhead is generated by Motif calls in displaying the windows and graphics. The memory overhead depends on the number of running processes, because **gpm** keeps information about all running processes in its user memory.

## REFERENCES

GlancePlus man pages: `gpm(1)`, `glance(1)`.

*Gpm Online Help User's Guide.*

*Glance Online Help User's Guide.*

*GlancePlus Adviser Manual, /usr/lpp/perf/paperdocs/gp/C/  
adviser.ps.Z.*

*GlancePlus Concepts Manual, /usr/lpp/perf/paperdocs/gp/C/  
concepts.ps.Z.*

*Performance Metrics Definition Manual, /usr/lpp/perf/paperdocs/  
gp/C/metrics.ps.Z.*

---

*Alex Polak  
System Engineer  
APS (Israel)*

© Xephon 2001

---

## **Free weekly Enterprise IS News**

A weekly enterprise-oriented news service is available free from Xephon. Each week, subscribers receive an e-mail listing around 40 news items, with links to the full articles on our Web site. The articles are copyrighted by Xephon – they are not syndicated, and are not available from other sources.

To subscribe to this newsletter, send an e-mail to [news-list-request@xephon.com](mailto:news-list-request@xephon.com), with the word subscribe in the body of the message. You can also subscribe to this and other Xephon e-mail newsletters by visiting Xephon's home page:

<http://www.xephon.com>

which contains a simple subscription form.

# Smit tutorial script

## INTRODUCTION

smitt.sh (smit tutorial) is a shell script that helps people to learn AIX commands while using smit. The script builds on the fact that commands within smit can be built and not executed. Smitt.sh runs smit in non-execution mode (ie builds the command but does not execute it), and allows users to build commands for various tasks and examine these later.

The script generates a single task log file (the log file contains commands from a single smit option) as well as an overall log file (the log file contains commands from all the smit options that have been selected so far).

However, the script is not complete, in the sense that it does not have all the options that are available in smit. But the script can easily be expanded to accommodate options that are currently not included.

## BUILDING NEW OPTIONS IN SMITT.SH

To build a new option in smitt.sh, you must do the following:

- 1 Add an entry for the top level menu option in the function DisplayfatPathRootMenu.
- 2 Build a new function that displays all the tasks that are available under this top level option.

Process these options as already done in the script.

## LISTING OF SMITT.SH

```
#####  
# Name      : smitt (smit tutorial)  
# Overview : The script allows the user to use the SMIT tool as a  
#           command learning tool. The script does not allow smit  
#           to execute any commands because smit is being run  
#           with the -x option. After each task is performed in  
#           simulation, the log file is displayed for the user to
```

```

#           view the details of the associated command(s). The
#           script creates a single task, and cumulative log
#           files which can be viewed or printed at any time.
#
# Notes    : 1. The script contains the following functions:
#           o main
#           o InitialiseVariables
#           o ProcessExit
#           o InitialiseLogFiles
#           o DisplayFastPathRootMenu
#           o StartSMITInNonExecutionMode
#           o DisplaySystemMenu
#           o DisplayPerformanceMenu
#           o DisplayTcipMenu
#           o DisplayHostMenu
#           o DisplayRouteMenu
#####
# Name      : InitialiseVariables
# Overview  : The function initializes all the variables.
# Notes    :
#####
InitialiseVariables ()
{
# define files
LAST_SCRIPT_FILE="/tmp/smit_$$_last.script"
ALL_SCRIPT_FILE="/tmp/smit_$$_all.script"
LAST_LOG_FILE="/tmp/smit_$$_last.log"
ALL_LOG_FILE="/tmp/smit_$$_all.log"
TEMP_FILE_1="/tmp/smit_$$_1.tmp"
TEMP_FILE_2="/tmp/smit_$$_2.tmp"
# define return codes
TRUE=0
FALSE=1
SEC=0
FEC=1
# define escape sequences
ESC="\0033["
RVON=_[7m           # reverse video on
RVOFF=_[27m        # reverse video off
BOLDON=_[1m         # bold on
BOLDOFF=_[22m       # bold off
BON=_[5m            # blinking on
BOFF=_[25m          # blinking off
# define required variables
SLEEP_DURATION=3    # no of seconds allowed for sleep command
ERROR="${RVON}${BON}smitt.sh:ERROR:${BOFF}"
INFO="${RVON}smitt.sh:INFO: "
# define menu title
SYSTEM_MENU="${RVON}System Maintenance Menu${RVOFF}"

```



```

TCPIP_MENU="${RVON}TCPIP Configuration Menu${RVOFF}"
HOST_MENU="${RVON}Host Configuration Menu${RVOFF}"
ROUTE_MENU="${RVON}Routing Configuration Menu${RVOFF}"
SERVICES_MENU="${RVON}Services Configuration Menu${RVOFF}"
ROOT_MENU="${RVON}Main Fast Path Menu${RVOFF}"
LANG_MENU="${RVON}Language Environment Menu${RVOFF}"
PERFORMANCE_MENU="${RVON}Main Fast Path Menu${RVOFF}"
# message
INTERRUPT="Program interrupted ! Quitting early${RVOFF}"
WORKING="Working.....${RVOFF}"
INVALID_ENTRY="Invalid entry${RVOFF}"
PRINT_OK="Successfully submitted print job${RVOFF}"
PRINT_NOT_OK="Failed to submit print job${RVOFF}"
OS_ERROR="\${SYSERROR}${RVOFF}"
NOT_INITIALISED="Failed to initialize existing file, \${FILE}${RVOFF}"
# define signals
SIGEXIT=0 ; export SIGEXIT # normal exit
SIGHUP=1 ; export SIGHUP # when session disconnected
SIGINT=2 ; export SIGINT # ctrl-c
SIGTERM=15 ; export SIGTERM # kill command
SIGTSTP=18 ; export SIGTSTP # ctrl-z command
}
#####
# Name : HandleInterrupt
# Overview : The function calls ProcessExit.
# Input :
# Notes :
#####
HandleInterrupt ()
{
DisplayMessage I "${INTERRUPT}"
ProcessExit $FEC
}
#####
# Name : MoveCursor
# Input : Y and X coordinates
# Returns : None
# Overview : It moves the cursor to the required location (Y,X).
# Notes :
#####
MoveCursor ( )
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
YCOR=$1
XCOR=$2
echo "${ESC}${YCOR};${XCOR}H"
}
#####
# Name : DisplayMessage
# Overview : The function displays message

```

```

# Input      : 1. Message type (E = Error, I = Informative)
#             2. Error Code as defined in DefineMessages ().
# Notes      :
#####
DisplayMessage ( )
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
MESSAGE_TYPE=$1
MESSAGE_TEXT=`eval echo $2`
clear
MoveCursor 24 1
if [ "${MESSAGE_TYPE}" = "E" ]
then
    echo "`eval echo ${ERROR}`${MESSAGE_TEXT}\c"
else
    echo "`eval echo ${INFO}`${MESSAGE_TEXT}\c"
fi
sleep ${SLEEP_DURATION}
return ${TRUE}
}
#####
# Name       : ProcessExit
# Overview   : The function processes a graceful exit.
# Input      : Exit Code
# Notes      :
#####
ProcessExit ( )
{
EXIT_CODE="$1"
rm -f ${LAST_SCRIPT_FILE}
rm -f ${ALL_SCRIPT_FILE}
rm -f ${LAST_LOG_FILE}
rm -f ${ALL_LOG_FILE}
rm -f ${TEMP_FILE_1}
rm -f ${TEMP_FILE_2}
exit $EXIT_CODE
}
#####
# Name       : InitialiseLogFiles
# Overview   : The function initializes log and script files.
# Returns    : $TRUE or $FALSE
# Notes      :
#####
InitialiseLogFiles ( )
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
# check the existence of script file
if [ -f ${LAST_SCRIPT_FILE} ]
then
    ( > ${LAST_SCRIPT_FILE} ) > /dev/null 2>&1
}

```

```

    if [ $? -ne 0 ]
    then
        FILE="${LAST_SCRIPT_FILE}"
        DisplayMessage E "${NOT_INITIALISED}"
        return $FALSE
    fi
fi
if [ -f ${ALL_SCRIPT_FILE} ]
then
    ( > ${ALL_SCRIPT_FILE} ) > /dev/null 2>&1
    if [ $? -ne 0 ]
    then
        FILE="${ALL_SCRIPT_FILE}"
        DisplayMessage E "${NOT_INITIALISED}"
        return $FALSE
    fi
fi
if [ -f ${LAST_LOG_FILE} ]
then
    ( > ${LAST_LOG_FILE} ) > /dev/null 2>&1
    if [ $? -ne 0 ]
    then
        FILE="${LAST_LOG_FILE}"
        DisplayMessage E "${NOT_INITIALISED}"
        return $FALSE
    fi
fi
if [ -f ${ALL_LOG_FILE} ]
then
    ( > ${ALL_LOG_FILE} ) > /dev/null 2>&1
    if [ $? -ne 0 ]
    then
        FILE="${ALL_LOG_FILE}"
        DisplayMessage E "${NOT_INITIALISED}"
        return $FALSE
    fi
fi
}
#####
# Name      : StartSMITInNonExecutionMode
# Overview  : The function starts SMIT tool in non-execution mode
#            using the fast path approach.
# Input     : Fast Path Name
# Notes     : 1. The following files are written into:
#            o LAST_SCRIPT_FILE (contains command details for
#            the last task)
#            o ALL_SCRIPT_FILE  (contains command details for all
#            tasks performed so far)
#            o LAST_LOG_FILE    (contains log details for the last
#            task)

```

```

#           o ALL_LOG_FILE      (contains log details for all
#                               tasks performed so far)
#####
StartSMITInNonExecutionMode ()
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
# assign parameter
FAST_PATH_PARAM="$1"
# start smit in non-execution mode
# Write additional command (not available in smit script file)
# for specific fast path parameter
if [ "${FAST_PATH_PARAM}" = "chgsys" ]
then
    echo "Display current system characteristics"    >>  ${TEMP_FILE_1}
    echo "lsattr -l sys0 -E \n "                    >>  ${TEMP_FILE_1}
fi
# write this particular task details
echo "${TASK}" >> ${TEMP_FILE_1}
smit -s ${TEMP_FILE_1} -l ${TEMP_FILE_2} -x ${FAST_PATH_PARAM}
# view script file
cp ${TEMP_FILE_1} ${LAST_SCRIPT_FILE}
cp ${TEMP_FILE_2} ${LAST_LOG_FILE}
view ${LAST_SCRIPT_FILE}
view ${LAST_LOG_FILE}
cat ${LAST_SCRIPT_FILE} >>  ${ALL_SCRIPT_FILE}
cat ${LAST_LOG_FILE} >>  ${ALL_LOG_FILE}
> ${TEMP_FILE_1}
> ${TEMP_FILE_2}
}
#####
# Name      : DisplayLanguageMenu
# Overview  : The function displays a menu with system-related tasks
# Input     : Fast Path Name
# Notes     :
#####
DisplayLanguageMenu ()
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
while true
do
clear
echo ""
echo " \t\t##### "
echo " \t\t# # "
echo " \t\t#  ${LANG_MENU} # "
echo " \t\t# # "
echo " \t\t#  5. Set User Language # "
echo " \t\t#  10. Change Language Hierarchy # "
echo " \t\t# # "
echo " \t\t#  98. Fast Path Root Menu # "

```

```

echo " \t\t# 99. Exit # "
echo " \t\t# # "
echo " \t\t##### "
echo "\t\t Enter Option(h for hot key) ---->\c"
read OPTION
case $OPTION in
    5 ) TASK="To set user language" ;
        StartSMITInNonExecutionMode "chlang_user" ;;
    10 ) TASK="To change language hierarchy" ;
        StartSMITInNonExecutionMode "mle_hier_cmd_hdr" ;;
    98 ) DisplayFastPathRootMenu ;
        break ;;
    99 ) ProcessExit $SEC ;;
esac
done
}
#####
# Name : DisplaySystemMenu
# Overview : The function displays a menu with system-related tasks
# Input : Fast Path Name
# Notes :
#####
DisplaySystemMenu ()
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
while true
do
clear
echo ""
echo " \t\t##### "
echo " \t\t# # "
echo " \t\t# ${SYSTEM_MENU} # "
echo " \t\t# # "
echo " \t\t# 5. Stop the System(shutdown) # "
echo " \t\t# 10. Assign the Console (chcons) # "
echo " \t\t# 15. Change/Show Date (date) # "
echo " \t\t# 20. Change/Show Time Zone (chtz) # "
echo " \t\t# 25. Manage Language Environment # "
echo " \t\t# 30. Change Operating System # "
echo " \t\t# Characteristics # "
echo " \t\t# 35. Broadcast Message to Users # "
echo " \t\t# 40. Run System Error Report # "
echo " \t\t# 45. System Dump # "
echo " \t\t# # "
echo " \t\t# 98. Fast Path Root Menu # "
echo " \t\t# 99. Exit # "
echo " \t\t# # "
echo " \t\t##### "
echo "\t\t Enter Option(h for hot key) ---->\c"
read OPTION
case $OPTION in

```

```

    5 ) TASK="To shutdown system" ;
        StartSMITInNonExecutionMode "shutdown" ;;
    10 ) TASK="To assign the console" ;
        StartSMITInNonExecutionMode "chcons" ;;
    15 ) TASK="To change the date" ;
        StartSMITInNonExecutionMode "date" ;;
    20 ) TASK="To change the time zone" ;
        StartSMITInNonExecutionMode "chtz" ;;
    25 ) DisplayLanguageMenu ;
        break ;;
    30 ) TASK="To change/show operating system characteristics" ;
        StartSMITInNonExecutionMode "chgsys" ;;
    35 ) TASK="Broadcast Message to All Users" ;
        StartSMITInNonExecutionMode "wall" ;;
    40 ) TASK="Report System Errors" ;
        StartSMITInNonExecutionMode "errpt" ;;
    45 ) TASK="Dump System" ;
        StartSMITInNonExecutionMode "dump" ;;
    98 ) DisplayFastPathRootMenu ;
        break ;;
    99 ) ProcessExit $SEC ;;
esac
done
}
#####
# Name      : DisplayTcipMenu
# Overview  : The function displays a menu for TCP/IP-related
#            options.
# Notes    :
#####
DisplayTcipMenu ()
{
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
while true
do
clear
echo ""
echo " \t\t##### "
echo " \t\t# "
echo " \t\t#      ${TCPIP_MENU}      # "
echo " \t\t# "
echo " \t\t#  5. Host "
echo " \t\t# 10. Routing "
echo " \t\t# 15. Network Interface "
echo " \t\t# 20. Name Resolution "
echo " \t\t# 25. Services "
echo " \t\t# 30. Remote Access "
echo " \t\t# 35. Daemons "
echo " \t\t# "
echo " \t\t# "
echo " \t\t# 98. Fast Path Root Menu "

```

```

echo " \t\t# 99. Exit                                     # "
echo " \t\t#                                             # "
echo " \t\t##### "
echo "\t\t Enter Option(h for hot key) ---->\c"
read OPTION
case $OPTION in
    5 ) DisplayHostMenu;
        break ;;
    10 ) DisplayRouteMenu ;
        break ;;
    25 ) DisplayServicesMenu ;
        break ;;
    98 ) DisplayFastPathRootMenu ;
        break ;;
    99 ) ProcessExit $SEC ;;
esac
done
}
#####
# Name      : DisplayHostMenu
# Overview  : The function displays a menu for TCP/IP-related
#            options.
# Notes    :
#####
DisplayHostMenu ()
{
while true
do
clear
echo ""
echo " \t\t##### "
echo " \t\t#                                             # "
echo " \t\t#      ${HOST_MENU}                               # "
echo " \t\t#                                             # "
echo " \t\t#  5. Show Host Name (lshostname)                 # "
echo " \t\t# 10. Set Hostname (mkhostname)                   # "
echo " \t\t#                                             # "
echo " \t\t# 98. Fast Path Root Menu                         # "
echo " \t\t# 99. Exit                                         # "
echo " \t\t#                                             # "
echo " \t\t##### "
echo "\t\t Enter Option(h for hot key) ---->\c"
read OPTION
case $OPTION in
    5 ) TASK="Show hostname" ;
        StartSMITInNonExecutionMode "lshostname" ;;
    10 ) TASK="Set a hostname" ;
        StartSMITInNonExecutionMode "mkhostname" ;;
    98 ) DisplayFastPathRootMenu ;
        break ;;
    99 ) ProcessExit $SEC ;;
esac
done
}

```

```

        esac
done
}
#####
# Name      : DisplayRouteMenu
# Overview  : The function displays a menu for routing-related tasks.
# Notes    :
#####
DisplayRouteMenu ()
{
while true
do
    clear
    echo ""
    echo " \t\t##### "
    echo " \t\t# "
    echo " \t\t#      ${ROUTE_MENU}      # "
    echo " \t\t# "
    echo " \t\t#  5. List All Routes      (lsroute ) # "
    echo " \t\t# 10. Add a Static Route   (mkroute ) # "
    echo " \t\t# 15. Remove a Static Route (rmroute ) # "
    echo " \t\t# 20. Flush Routing Table  (fshrttbl) # "
    echo " \t\t# "
    echo " \t\t# 98. Fast Path Root Menu  # "
    echo " \t\t# 99. Exit                  # "
    echo " \t\t# "
    echo " \t\t##### "
    echo "\t\t Enter Option(h for hot key) ---->\c"
    read OPTION
    case $OPTION in
        5 ) TASK="List all routes" ;
            StartSMITInNonExecutionMode "lsroute" ;;
        10 ) TASK="Add a static route" ;
            StartSMITInNonExecutionMode "mkroute" ;;
        15 ) TASK="Remove a static route" ;
            StartSMITInNonExecutionMode "rmroute" ;;
        20 ) TASK="Flush routing table" ;
            StartSMITInNonExecutionMode "fshrttbl" ;;
        98 ) DisplayFastPathRootMenu ;
            break ;;
        99 ) ProcessExit $SEC ;;
    esac
done
}
#####
# Name      : DisplayServicesMenu
# Overview  : The function displays a menu for routing-related tasks.
# Notes    :
#####
DisplayServicesMenu ()
{

```



```

while true
do
  clear
  echo ""
  echo " \t\t##### "
  echo " \t\t# "
  echo " \t\t#      ${SERVICES_MENU}      # "
  echo " \t\t# "
  echo " \t\t#  5. List All Services      (lsaervices ) # "
  echo " \t\t# 10. Add a Service          (mkservices ) # "
  echo " \t\t# 15. Show Characteristics of a "
  echo " \t\t#           Service (shservices ) # "
  echo " \t\t# 20. Change Characteristics of a "
  echo " \t\t#           Service (chservices ) # "
  echo " \t\t# 25. Remove a Service      (rmservices) # "
  echo " \t\t# "
  echo " \t\t# 98. Fast Path Root Menu "
  echo " \t\t# 99. Exit "
  echo " \t\t# "
  echo " \t\t##### "
  echo "\t\t Enter Option(h for hot key) ---->\c"
  read OPTION
  case $OPTION in
    5 ) TASK="List all Services" ;
        StartSMITInNonExecutionMode "lsservices" ;;
    10 ) TASK="Add a Services" ;
        StartSMITInNonExecutionMode "mkservices" ;;
    15 ) TASK="Show Characteristics of a Service" ;
        StartSMITInNonExecutionMode "shservices" ;;
    20 ) TASK="Change Characteristics of a Service" ;
        StartSMITInNonExecutionMode "chservices" ;;
    25 ) TASK="Remove a Service" ;
        StartSMITInNonExecutionMode "rmservices" ;;
    98 ) DisplayFastPathRootMenu ;
        break ;;
    99 ) ProcessExit $SEC ;;
  esac
done
}
#####
# Name      : DisplayFastPathRootMenu ()
# Overview  : The function displays main menu.
# Notes     :
#####
DisplayFastPathRootMenu ()
{
  trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
  while true
  do
    clear

```

```

echo ""
echo " \t\t##### "
echo " \t\t# "
echo " \t\t#      ${ROOT_MENU}      # "
echo " \t\t# "
echo " \t\t# 5. Managing System Environment # "
echo " \t\t# 10. Managing System Performance # "
echo " \t\t# 15. Managing TCPIP # "
echo " \t\t# # "
echo " \t\t# # "
echo " \t\t# 85. View File # "
echo " \t\t# 90. Print File # "
echo " \t\t# # "
echo " \t\t# 99. Exit # "
echo " \t\t# # "
echo " \t\t##### "
echo "\t\t Enter Option(h for hot key) ---->\c"
read OPTION
case $OPTION in
    5 ) DisplaySystemMenu ;
        break ;;
    10 ) DisplayPerformanceMenu ;
        break ;;
    15 ) DisplayTcpiMenu ;
        break ;;
    99 ) ProcessExit $SEC ;;
    * ) DisplayMessage E "${INVALID_ENTRY}" ;;
esac
done
}
#####
# Name : main
# Overview : The function invokes all other functions.
# Notes :
#####
main ()
{
InitialiseVariables
if ! InitialiseLogFiles
then
ProcessExit $FEC
fi
DisplayFastPathRootMenu
}
# invoke main
main

```

---

*Arif Zaman*  
*Analyst/Programmer*  
*High-Tech Software (UK)*

© Xephon 2001

---

## Standby system failover

This article describes how to set up a production/standby system arrangement for managing remote sites from a centralized location without using the official HACMP software and hardware. It's a 'poor-man's HACMP', if you will.

HACMP (High Availability Cluster Multi Processing) is a proven product that provides the redundancy and fault resilience required for business-critical applications. However the implementation and maintenance of it is not a trivial matter. Furthermore it may be overkill for some business needs – trying to be all things to all people, when a more simple home-made solution might suffice. There are various reasons why one needs to switch to a standby machine, and they are not all cases of catastrophic failure. Perhaps it is simply the addition of new adapters or internal disks that requires the running of production on the standby machine for a period of time while the primary system is being worked on.

Using the methods described in this article presupposes that your business could withstand approximately fifteen minutes of downtime during the switchover. Obviously this may not be acceptable if you are in the financial business. However, there are many businesses that could tolerate a short outage while switching from the main production server to a hot standby machine in the (hopefully rare) event that the main system suffers a failure.

For AIX administrators who have to support remote facilities that have no local RS/6000 or AIX expertise on site, it is critical to set up an environment that is easy to manage from the centralized location with minimal assistance from the remote staff. It was decided that a system could be designed whereby a failed production machine could be switched over to a standby machine within a reasonable amount of time, while at the same time avoiding the cost and complexity of using the official HACMP product. The other prerequisite was that this could all be done remotely with little or no assistance from the staff at the remote location.

In general, when the production machine fails, a service call should immediately be placed to IBM. Once the failover to the standby

system is successfully accomplished, the users can log back in and carry on their work. The standby system will appear to be the production machine to the users. In the meantime, the responding technician from IBM can proceed to repair the real production machine. Later, at an appropriate time, after the production system has been repaired, the reverse switchover operation can be executed, where the standby machine will relinquish the production data to the normal production machine and resume its role as a standby server once again.

The basics of this involve using two RS/6000 systems, external SSA disk storage (which is cabled to both systems but owned by only one of the systems at a given time), switching hostnames and IP addresses (switching system identities), and having a dial-up modem to the production system, just in case. It is also important to keep all the production data in a separate volume group (or groups) on the external storage, apart from rootvg, since this data will be taken over by the standby machine.

Regarding rootvg, any software that is required on the production machine must also be installed on the standby machine (for example, DB2, SNA, etc). While the software may not normally actually be in use on the standby, it must be there for the day when the standby must takeover the production role.

If specific configuration files exist which occasionally change, they must be copied over nightly from production to standby. For example, */etc/users*, */etc/group*, */etc/security/users*, */etc/security/group*, etc, and any other relevant rootvg files. This may also include crontabs and various other config or init files that may be required by the applications running on the system. The goal is to keep the standby system up to date with the latest rootvg changes. For this purpose, refer to the included scripts called **box\_to\_box.sh** and **sync\_user\_info**. These two scripts run nightly on the standby machine and will keep it current, based on the primary machine's configuration and user settings.

The nightly copy job ensures that the user accounts (including the latest password changes) should all be up-to-date (within 24 hours) on the standby system. It is important to have a separate filesystem where all user home directories exist, instead of the normal */home* filesystem.

This separate filesystem can then be part of the external disk storage in a non-rootvg volume group and will then be available when the standby machine imports the non-rootvg VG(s).

The basic high-level steps during the failover are as follows:

- Determine that the situation warrants failing over to the standby machine (eg failed power supply in the production system, system crashed and will not reboot, hardware changes, etc).
- If necessary, have the remote users log off the system (they may have been forced off anyway, depending on the severity of the problem).
- If the primary system is still accessible, change its identity (alternate hostname and IP address).
- Log into the standby machine and have it takeover the SSA storage, change its identity (hostname and IP address) to that of the normal production server.
- Have the people log back in again and resume working (they should see no difference from the normal production system).
- Perform the required maintenance/repairs on the main system.
- Later, when circumstances allow, restore the systems back to their normal roles and configurations.

## PREREQUISITES

### **Hardware set-up prerequisites**

The sample remote site described in this article has a Model F50 production server and a Model 570 standby server. The Model 570 has significantly less processing power than the F50 system, but it is enough to get by while the F50 undergoes repairs. This is more cost-effective than having a more expensive second F50 standing by. Both servers have SSA adapters that are cabled to an external 7133-600 deskside SSA storage subsystem. Of course any IP addresses specified in this article will have to be modified to reflect your network addresses. As a convention, we will assume the primary system's IP address ends in 12 and the standby system's IP address ends in 13, and

that they are both on a network xx.xxx.114 (the mask is 255.255.255.0).

```
Production system:  hostname is newy01          IP address is xx.xxx.114.12
Standby system    :  hostname is newy01_570    IP address is xx.xxx.114.13
xx.xxx.114.12                                         xx.xxx.114.13
```

The sample system runs an application that requires an SNA connection through a router for communication with the home-office mainframe system. Therefore each system has a Multi-Protocol Adapter (MPA) which is cabled to an A/B switch box. When the F50 is in use, the box is set to position A. When the machine identities are switched, this is the one time we need a person at the remote location to physically assist, and change the A/B switch from A to B so that the standby Model 570's SNA connection is used in place of the F50's. Of course this A/B box and SNA set-up is only peculiar to this sample application server. It is *not* an integral part of the standby failover procedure.

### User account set-up prerequisites

Create a separate filesystem for user home directories since we want to put them on the external disks to be taken over along with the production data. We will call this filesystem */users*.

For existing users, I have provided a script (**move\_users.sh**) which will copy each user's existing */home* files into their */users/username* tree. This should ideally be done when the users are not logged in. It will also change the specification of their home directory to */users/username* (eg */users/smith*). In order to make the */users* filesystem the new default place to create new users' directories when adding new users via *smit*, edit the file */etc/security/mkuser.default*. In this file, change the two lines that point to */home* and substitute */users* instead. From then on, when a new user account is created, it will point the user's new home directory to */users* instead of the traditional */home* filesystem.

### PRODUCTION DATA VOLUME GROUP

The sample system has its production data in a volume group called *datavg* which resides on the external disk subsystem. There are several filesystems in this *datavg* volume group. If the F50 is in good enough shape to still login to it, the filesystems will all be dismounted

and the volume group varied offline and exported in order to make it available for takeover by the 570 machine. If the F50 is unavailable because of a serious problem, then it can be shutoff and its network cable can be disconnected, and thus has essentially already given up its ownership of the data.

## CHANGING HOSTNAMES AND IP ADDRESSES

Care must be taken not to allow both machines to have the same hostname or IP address at the same time! First we change the production system to an obscure address like xx.xxx.114.239 (choose an unused IP address for this temporary address) with a temporary hostname as well. Then we change the standby 570 system's identity to use the F50's normal hostname and IP address.

In order to change the hostname and IP, a small script is 'nohupped' to execute the changeover. Again, since this is done remotely, the current Telnet session will hang/disconnect as soon as the network interface is (software) detached. So it is essential to *nohup* the script so that it can fully complete in the background. Then the admin can log back in again to complete the final failover steps.

Note that it is helpful to have a dial-in modem attached to the systems just in case the TCP/IP address switching has a problem. Without this, a remote user would have to be coached through logging into the remote console and going into smitty tcpip. Normally this precaution is not necessary, but it provides an extra comfort zone just in case.

There are a total of four TCP/IP changeover scripts:

- **chg12to239** (F50 to temporary IP while it's being repaired)
- **chg13to12** (570 becomes F50)
- **chg12to13** (570 resumes being 570 again)
- **chg239to12** (F50 resumes being F50 again).

Note that the chg\* scripts above are all called from within the cutover and restore scripts listed below.

The following cutover scripts implement the failover to the standby machine:

- **F50cutover.sh**
- **570cutover.sh.**

The following restore scripts return the production and standby machines to their normal roles:

- **F50restore.sh**
- **570restore.sh.**

#### RESTARTABLE STEPS

The cutover and restore scripts are written in a way that permits re-running the scripts from any desired step by specifying that step as an argument.

#### ACTUAL SEQUENCE OF EVENTS

To accomplish the failover, here is what would be done, in chronological order:

- Implement the required prerequisites as listed above, including the disk cabling, the datavg volume (including the */users* filesystem) residing on the external disks. To move the user data to the */users* filesystem, use the **move\_users.sh** script once.
- Customize the included scripts to suit your environment, application, and hardware needs, including your IP addresses/ hostnames, filesystem names, vgnames, etc.
- Have the standby machine run the **box\_to\_box.sh** script nightly via cron at around 3am. This **box\_to\_box.sh** script calls the **sync\_user\_info** script.
- Have the following scripts in the '/' root directory of the primary machine:
  - **F50cutover.sh**



- **F50restore.sh**
- **ImportVG**
- **chg12to239**
- **chg239to12.**
- Have the following scripts in the ‘/’ root directory of the standby machine:
  - **570cutover.sh**
  - **570restore.sh**
  - **ImportVG**
  - **chg13to12**
  - **chg12to13.**
- Telnet to the remote site’s production machine:
  - Have the users logoff.
  - **/F50cutover.sh** (change production machine to temporary identity).
- Telnet to the remote site’s standby machine:
  - **/570cutover.sh** (make it become the production machine).
- Have the users login and resume normal work.

Later when it is time to set the systems back to their normal roles again:

- Telnet to the standby system using the production system’s IP address/hostname:
  - **/570restore.sh** (make it become the standby machine again).
- Telnet to the production machine using the temporary IP ‘239’ address:
  - **/F50restore.sh** (make it become the production machine again).

## SYNC\_USER\_INFO

```
# sync_user_info
# This shell will run nightly, called by the "boxtobox.sh" script, and
# will sync the user files (user, group, passwd) from the production
# machine with the standby machine.
#=====
echo "$(basename $0): $(date) - Processing starting."
#...Set the "live_node" to the nodename of the live RS/6000
#...This assumes that the standby system's hostname is the same
#...as the live system's except that it ends in "_570".
#...Used sed to strip off everything from the underscore on...
live_node="$(hostname | sed -e 's/_.*$//')"
echo "$(basename $0): $(date) - Files will be copied from ${live_node}."
for file in /etc/security/user /etc/group /etc/security/group /etc/
passwd /etc/security/passwd
do
    echo "$(basename $0): $(date) - Copying the ${file} file ..."
#...For rcp to work, the production system's root .rhosts file needs to
#...allow the standby system's root account trusted access...
    /usr/bin/rcp -p ${live_node}:${file} ${file}
done
# end of job
echo "$(basename $0): $(date) - Processing complete."
```

## BOX\_TO\_BOX.SH

```
# boxtobox.sh
# Runs nightly from cron, and copies the production system's cron
# files over to the standby system. It also calls sync_user_info which
# brings over the user account files to keep the user information
# current.
#-----
#...Set the "live_node" to the nodename of the live RS/6000
#...This assumes that the standby system's hostname is the same
#...as the live system's except that it ends in "_570".
#...Use sed to strip off everything from the underscore on...
live_node="$(hostname | sed -e 's/_.*$//')"
echo "$(basename $0): $(date) - Copying over production cronfiles from
${live_node}... "
cd /var/spool/cron/crontabs
#..Note that the rcp command will only work if the production system's
#..root account permits trusted access to the standby system in its
#../.rhosts file...
/usr/bin/rcp -pr ${live_node}:/var/spool/cron/crontabs/root/ ./F50_root
/usr/bin/rcp -pr ${live_node}:/var/spool/cron/crontabs/some_account/ ./
F50_some_account
date
#...Call the user account synchronization script...
/sync_user_info
```

```
echo "$(basename $0): $(date) - Processing complete."  
# end of job
```

## CHG12TO13

```
#!/bin/ksh  
# chg12to13  
# This script should be stored in the / root directory.  
# Change TCP/IP hostname and address of standby machine back to its  
# normal settings.  
# Note: all 'xx' should of course be edited to supply your appropriate  
#       IP addresses for hostname, gateway (-g) and nameserver (-n).  
#       The assumed Ethernet interface is "en0". If this is different  
#       in your situation, substitute the appropriate interface name.  
#...detach the en0 interface in order to make the configuration  
# change...  
ifconfig en0 detach  
  
/usr/sbin/mktcpip -h'newy01_570' -a'xx.xxx.114.13' -m'255.255.255.0' -  
i'en0' -n'xx.xxx.xx.xx' -d'your.company.domain.com' -g'xx.xxx.114.1' -  
t'N/A'  
  
#...bring the interface back up again  
ifconfig en0 up
```

## CHG12TO239

```
#!/bin/ksh  
# chg12to239  
# This script should be stored in the / root directory.  
# Change TCP/IP hostname and address of primary machine to a temporary  
# hostname and IP address while it will be undergoing repairs.  
# Note: all 'xx' should of course be edited to supply your appropriate  
#       IP addresses for hostname, gateway (-g) and nameserver (-n).  
#       The assumed ethernet interface is "en0". If this is different  
#       in your situation, substitute the appropriate interface name.  
#...detach the en0 interface in order to make the configuration  
# change...  
ifconfig en0 detach  
/usr/sbin/mktcpip -h'newy01_temp' -a'xx.xxx.114.239' -m'255.255.255.0' -  
i'en0' -n'xx.xxx.xx.xx' -d'your.company.domain.com' -g'xx.xxx.114.1' -  
t'N/A'  
#...bring the interface back up again  
ifconfig en0 up
```

## CHG13TO12

```
#!/bin/ksh  
# chg13to12
```

```

# This script should be stored in the / root directory.
# Change TCP/IP hostname and address of standby machine to make it take
# on the identity of the primary system.
# Note: all 'xx' should of course be edited to supply your appropriate
#       IP addresses for hostname, gateway (-g) and nameserver (-n).
#       The assumed ethernet interface is "en0". If this is different
#       in your situation, substitute the appropriate interface name.
#...detach the en0 interface in order to make the configuration
# change...
ifconfig en0 detach
/usr/sbin/mktcpip -h'newy01' -a'xx.xxx.114.12' -m'255.255.255.0' -i'en0'
-n'xx.xxx.xx.xx' -d'your.company.domain.com' -g'xx.xxx.114.1' -t'N/A'
#...bring the interface back up again
ifconfig en0 up

```

## CHG239TO12

```

#!/bin/ksh
# chg239to12
# This script should be stored in the / root directory.
# Change TCP/IP hostname and address of primary machine back to its
# normal settings.
# Note: all 'xx' should of course be edited to supply your appropriate
#       IP addresses for hostname, gateway (-g) and nameserver (-n).
#       The assumed ethernet interface is "en0". If this is different
#       in your situation, substitute the appropriate interface name.
#...detach the en0 interface in order to make the configuration
# change...
ifconfig en0 detach

/usr/sbin/mktcpip -h'newy01' -a'xx.xxx.114.12' -m'255.255.255.0' -i'en0'
-n'xx.xxx.xx.xx' -d'your.company.domain.com' -g'xx.xxx.114.1' -t'N/A'
#...bring the interface back up again
ifconfig en0 up

```

## IMPORTVG

```

#!/bin/ksh
# /ImportVG
# This script will import the datavg volume group, vary it on,
# and perform fsck (filesystem-check) on the filesystems before
# mounting them.
# Note that it is best to store this ImportVG script right in the /
# root directory, along with the /FSnames.dat file (see below).
#...Change the name of the VG as well as the diskname in the following
#...lsvg command to the appropriate values for your system...
set -v
lsvg |grep datavg || importvg -y datavg hdisk2
varyonvg datavg

```

```

# Doing filesystem check as well as setting automount to true,
# for all the concerned filesystems.
# This optional step is good insurance that all the filesystems
# involved are in optimum shape. In order to use this, you must
# have previously created a file in the "/" directory called
# /FS_names.dat. This file will contain all the filesystems that
# are in the volume group being imported. One by one, they are
# set to 'automount=true' and fsck is run on them.
# The /FSnames.dat file would be set up in this format:
#/filesystem1
#/filesystem2
#/filesystem3
#...etc.
while read FS
do
    grep -q ${FS} /etc/filesystems
    if [ $? -eq 0 ]
    then
        chfs -A yes ${FS}
        fsck ${FS}
    fi
done < /FS_names.dat
#...Now do the mount command to mount all the filesystems, and show the
#...status of the filesystems with the df command.
mount all
df -kv
set +v

```

## MOVE\_USERS.SH

```

# move_users
# This shell will move all the files from the users' /home directory
# into the "/users" filesystem. It will act upon any and all users who
# belong to the user-groups listed in the 'for group in' loop. Add any
# groups whose users you wish to move out of /home into /users.
#=====
echo "$(basename $0): $(date) - Processing starting."
# Make sure the /users filesystem exists
lsfs /users 2>/dev/null
if [ $? -ne 0 ]
then
    echo "$(basename $0): $(date) - *** The /users filesystem does not
exist."
    echo "$(basename $0): $(date) - *** Processing terminated."
    exit 1
fi
#...For this next step, you can specify any user groups you are
#...interested in moving....
for group in staff somegroup      # <--- specify any desired user groups
here

```

```

do
    lsgroup ${group} >/dev/null 2>/dev/null
    if [[ $? -ne 0 ]]
    then
        echo "Group ${group} does not exist. Skipping..."
        continue
    fi
    echo "\tProcessing group ${group} ..."
#...Make a list of all the users in this group (eg user1 user2 user3)
    for user in $(lsgroup -f -a users ${group} |grep users= | sed -e 's/
    ..*=/ /' -e 's/\,/ /g')
        #strip off 'users=' and substitute commas w/spaces..
        do
#...Get the home directory of this user, taking everything to the right
#...of the '=' sign.
            home_dir="$(lsuser -a home ${user} | sed -e 's/..*=/ /')"
#...If the user's home directory is neither /u nor /home,
# skip that user..
            if [ "$(echo ${home_dir} | cut -c1-3)" != "/u/" -a "$(echo
            ${home_dir} | cut -c1-5)" != "/home" ]
            then
                echo "Skipping user ${user} since homedir is ${home_dir}..."
                continue #go to next iteration of the for loop
            fi
            echo "\t\tProcessing user ${user} ..."
            chuser home=/users/${user} ${user} #set the new home dir
            if [ -d ${home_dir} ]
            then
                mv ${home_dir} /users #rename the directory
            else
                echo "\t\t\tUser ${user}'s previous home directory did not exist."
                echo "User's home directory has been updated."
            fi
        done
    done
done
echo "$(basename $0): $(date) - Processing complete."
# end of job

```

## F50RESTORE.SH

```

#!/bin/ksh
set -v
# F50restore.sh
#=====
# DESCRIPTION:
# -----
# This will bring the F50 system back to its normal settings. #
#
# INSTRUCTIONS FOR USE:
# -----

```

```

# This should be run after running 570restore.sh which puts #
# the 570 standby system back to its normal standby role. #
# #
# Telnet to the remote production F50 system now that it is #
# ready to once again become the live system. #
# Then from the root account, execute the following command: #
# /F50restore.sh #
# #
# Once the ip-address gets changed in the step "CHANGE_IP", #
# you will be disconnected from the remote site. You must then#
# reconnect using the xx.xxx.xxx.12 ip_address. #
# #
# Once reconnected, execute this script passing #
# the step-argument as follows: #
# /F50restore.sh IMPORT_VG #
# #
# This will resume the processing after the ip-change. #
#=====#
date
if [ $1 != "" ] #if no arguments, assume step 1
then
STEP=$1
else
STEP="SWITCH_AB"
fi
#=====#
date
#...Now have the remote site person switch the A/B box back to the
#..."A" (F50) position for the purpose of SNA communication with the
#...remote mainframe system which this machine's application requires.
if [ ${STEP} = "SWITCH_AB" ]
then
set +v
echo "Please have the remote site person switch the A/B box to the
F50 position now."
echo "\nHas this has been done? (y/n) \c"
read answer extra
case ${answer} in
"y"|"Y")
echo "Proceeding to switch the IP address..."
;;
*)
echo "Exiting procedure. Please rerun this when you are"
echo "prepared to switch the A/B box."
exit 1
;;
esac
STEP="CHANGE_IP"
fi
#=====#
date
set -v

```

```

#...Change the hostname and IP address back to the production
#...values. It is ESSENTIAL to "nohup" this step in the background
#...( & ) in order to allow it to complete, as it will disconnect
#...the telnet session as the ethernet interface is detached...
if [ ${STEP} = "CHANGE_IP" ]
then
    set +v
    echo "Detaching the network interface..you will be disconnected.."
    echo "In a few moments, please login to the production IP address"
    echo "and execute the following command:"
    echo "/F50restore.sh  IMPORT_VG"
    echo ""
    nohup ksh /chg239to12  &
    sleep 10
    exit 0      #exit out of the script. Then telnet back in...
    STEP="IMPORT_VG"
fi
#=====
date
#...Import the production datavg volume group back into the
#...system again, and start up the SNA sessions for this
#...system's application requirements.
if [ ${STEP} = "IMPORT_VG" ]
then
    set -v
    /ImportVG
    STEP="START_SNA"
fi
#=====
date
if [ ${STEP} = "START_SNA" ]
then
    /MBSYSMGR/restart_sna
    STEP="CRONTAB"
fi
#=====
date
#...Restore the production crontabs and refresh the daemon...
if [ ${STEP} = "CRONTAB" ]
then
    mv /var/spool/cron/crontabs/root_save      \
        /var/spool/cron/crontabs/root
    mv /var/spool/cron/crontabs/some_account_save      \
        /var/spool/cron/crontabs/some_account
    pidcrond=$(ps -aef | grep -v grep | grep "cron" |awk '{print $2}')
    kill -9 ${pidcrond}
fi
#=====
date
set +v
echo "The F50restore procedure is now COMPLETE."
#(end of job).

```



## 570RESTORE.SH

```
#!/bin/ksh
set -v
# 570restore.sh
#=====
# DESCRIPTION:                                     #
# -----                                         #
# This script will put the 570 standby system back to normal. #
# Its hostname will revert back to <hostname>_570, and its   #
# ip_address will become xx.xxx.xxx.13 again.                #
#                                                           #
# INSTRUCTIONS FOR USE:                               #
# -----                                         #
# This script will be used when the standby system is ready #
# to hand back the production data to the real production   #
# machine which has been repaired.                        #
#                                                           #
# Contact the remote site and have the users logout.        #
# Telnet to the standby 570 (which still appears to be the  #
# production system with the production hostname and IP addr) #
# and then execute this script using the root account, as   #
# follows:                                                  #
#     /570restore.sh                                       #
#                                                           #
# NOTE:                                                  #
# ----                                                  #
# Once the ip-address gets changed in the step "CHANGE_IP", #
# you will be disconnected from the site. At this point this #
# procedure is done. Just to verify that it did work, you   #
# can reconnect to the xx.xxx.xxx.13 address and verify it. #
#=====
date
if [[ $1 != "" ]]      #if no argument, assume step 1
then
    STEP=$1
else
    STEP="SHUTDOWN_SNA"
fi
#=====
date
#...In this step we shutdown the SNA sessions that are running
#...on this production system which communicates with a remote
#...mainframe system.
#...The standby system does not require the connections...
if [ ${STEP} = "SHUTDOWN_SNA" ]
then
    /SYSMGR/sna_shutdown
    STEP="RENAME_CRONS"
fi
#=====
```

```

date
#...In this step we restore the 570's normal cron jobs (if any) since
#...this standby system will no longer need to run the production
#...crons...
if [ ${STEP} = "RENAME_CRONS" ]
then
    cp /var/spool/cron/crontabs/570_root      \
      /var/spool/cron/crontabs/root
    cp /var/spool/cron/crontabs/570_some_account \
      /var/spool/cron/crontabs/some_account
#...Now refresh the cron daemon...
pidcrond=$(ps -aef | grep -v grep | grep "cron" |awk '{print $2}')
kill -9 $pidcrond
STEP="KILL_MISC"
fi
#=====#
date
#...In this step we are killing any known running jobs that would
#...be holding open any files in filesystems that we later intend
#...to dismount. In other words, we want to avoid the message that
#...says 'filesystem is busy' when we try to unmount them.
if [ ${STEP} = "KILL_MISC" ]
then
    fuser -k /dev/some_app_LV      # -k should kill all the PIDs found
    umount /some_app_filesystem  #(see the associated LV above)
    if [ $? -ne 0 ]
    then
        set +v
        echo "Could not unmount the /some_app_filesystem filesystem."
        echo "Please fix this and restart from step KILL_MISC"
        echo "i.e. /570restore.sh  KILL_MISC"
        echo "Terminating procedure..."
        exit 1
    fi
    STEP="UNMOUNT_ALL"
fi
#=====#
date
#...Now unmount all the filesystems (except for the normal system
#...filesystems such as /, /usr, etc...
if [ ${STEP} = "UNMOUNT_ALL" ]
then
    umount -a
    STEP="VARY_OFF"
fi
#=====#
date
#...Vary off the volume group
if [ ${STEP} = "VARY_OFF" ]
then

```

```

varyoffvg datavg
if [ $? -ne 0 ]
then
    set +v
    echo "Could not varyoff VG datavg."
    echo "Please fix this and restart from step VARY_OFF"
    echo "i.e. /570restore.sh VARY_OFF"
    echo "Terminating procedure..."
    exit 1
fi
STEP="EXPORT_VG"
fi
#####
date
#...Relinquish the production data volume group
if [ ${STEP} = "EXPORT_VG" ]
then
    exportvg datavg
    if [ $? -ne 0 ]
    then
        set +v
        echo "Could not export VG ivisvg."
        echo "Please fix this and restart from step EXPORT_VG"
        echo "i.e. /570restore.sh EXPORT_VG"
        echo "Terminating procedure..."
        exit 1
    fi
    STEP="CHANGE_IP"
fi
#####
date
#...Resume using the standby system's normal hostname and
#...IP address.
set +v
echo "Detaching the network interface...you will be disconnected.."
echo "(No further action on this machine is necessary.)"
if [ ${STEP} = "CHANGE_IP" ]
then
    nohup ksh /chg12to13 &
    sleep 10
    exit 0
fi
#####
#end of job.

```

*Editor's note: this article will be concluded in the next issue.*

---

*Michael G Stanton  
Supervisor Mid-range Systems  
Mercedes-Benz (USA)*

© Xephon 2001

---

# AIX news

---

Searchspace has announced support for AIX with its Intelligent Enterprise Framework for automating enterprises using intelligent systems that interpret activity and initiate action when risks and opportunities arise.

It can apparently be used to detect behaviour ranging from money laundering and payment fraud to customer loyalty analysis and cross-selling opportunities. Systems such as accounting, billing, CRM, and data marts can be integrated into the framework.

For further information contact:  
Searchspace, Prospect House, New Oxford St, London WC1A 1HB, UK.  
Tel: (020) 7255 1065.  
URL: <http://www.searchspace.com/news/press16022001.shtml>.

\* \* \*

Simulus has unveiled the IBM TSSP for SIMperformer, which supports the NUMA-Q 9000 Model E410 server – up to two quads using DL2 interconnect or up to 16 quads using the IQLink interconnect – including support for Pentium Pro, Pentium II, Xeon, and Pentium III Xeon chips.

Also supported are the F80, H80, and M89 RS/6000 servers, including the PowerPC RS64 III 450MHz and 500MHz processors. I/O subsystem support covers the Enterprise Storage Server 2105 F10 and F20, the ExpPlus 2104 DLI and TL1, the IBM 7133 D40 and T40, and the Fibre Channel RAID server.

For further information contact:  
Simulus Ltd, Corinthian Court, 80 Milton Park, Abingdon, Oxon OX14 4RY, UK.

Tel: (01235) 827400.  
URL: <http://www.simulus.com/prodsimperformer.htm>.

\* \* \*

RealNetworks has begun shipping its RealSystem iQ Server and Proxy on AIX, HP-UX, and Solaris.

RealSystem iQ provides a framework for global Internet media delivery from any platform, in any format, and to any device. It creates, says the firm, a peer-to-peer network of streaming servers optimized for the cost-effective and reliable delivery of digital media.

It operates at multiple levels to eliminate congestion in the middle of the network, avoiding the need for servers to distribute media streams first to edge servers and then to consumers.

For further information contact:  
RealNetworks, PO Box 91123, Seattle, WA 98111-9223, USA.  
Tel: (206) 674 2700.  
URL: <http://www.realnetworks.com/iq/>.

\* \* \*

IBM has announced Version 1.1 of its WebSphere Portal Server for AIX, which provides a framework for enabling the aggregation of information from various content sources and applications into a personalized portal.

For further information contact your local IBM representative.



**xephon**