

101

AIX

March 2004

In this issue

- 3 Hardware Management Console: using remote commands with secure shell from AIX
 - 3 The power of awk
 - 13 LPAR memory management
 - 27 Using HMC command line interface to manage Logical Partitioning Facility (LPAR)
 - 53 AIX news
-

© Xephon Inc 2004

update

AIX Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Nicole Thomas
E-mail: nicole@xephon.com

Subscriptions and back-issues

A year's subscription to *AIX Update*, comprising twelve monthly issues, costs \$275.00 in the USA and Canada; £180.00 in the UK; £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 2000 issue, are available separately to subscribers for \$24.00 (£16.00) each including postage.

***AIX Update* on-line**

Code from *AIX Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at <http://www.xephon.com/aix>; you will need to supply a word from the printed issue.

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 \$160 (£100 outside North America) per 1000 words and \$80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of \$32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

© Xephon Inc 2004. 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.

Printed in England.

The power of awk

awk is a powerful command line utility that comes with virtually all Unixes, including AIX.

awk receives two inputs: the data file (often piped into it from another command) and the command file. In the first of this series of snippets on **awk**'s power I'd like to illustrate a time-saving feature of its use.

For example, if you have a directory of file names that was brought over, say, from Windows, and the filenames are all in caps, assuming the directory is current:

```
/tmp/awkexample # ls  
FILEA FILEB FILEC  
/tmp/awkexample #
```

```
/tmp/awkexample # ls | awk '{print "mv " $1 " " " tolower($1)}'  
mv FILEA filea  
mv FILEB fileb  
mv FILEC filec
```

you could then redirect this output to a file, edit it (if need be), and then run it. Obviously this is a quick example; if there were hundreds of files, the time saving would be significant.

David Miller
Database Architect
Baystate Health Systems (USA)

© Xephon 2004

Hardware Management Console: using remote commands with secure shell from AIX

A number of recent AIX servers, such as Bull's PL3200 and IBM's p690, can be divided into two or more discrete partitions. Each partition has its own resources (processors (CPUs), memory, and/or I/O slots) and runs its own operating system and

applications. Furthermore it is possible to move resources from one partition to another without requiring a shutdown and reboot of the partitions involved. This capability is known as dynamic logical partitioning, or DLPAR.

The application that manages DLPAR and the resource allocation to each partition runs on a standard Intel-based PC running Red Hat Linux. This PC is called the Hardware Management Console (HMC). The HMC is connected to the server hypervisor by an RS232 serial line and to each partition by Ethernet (see

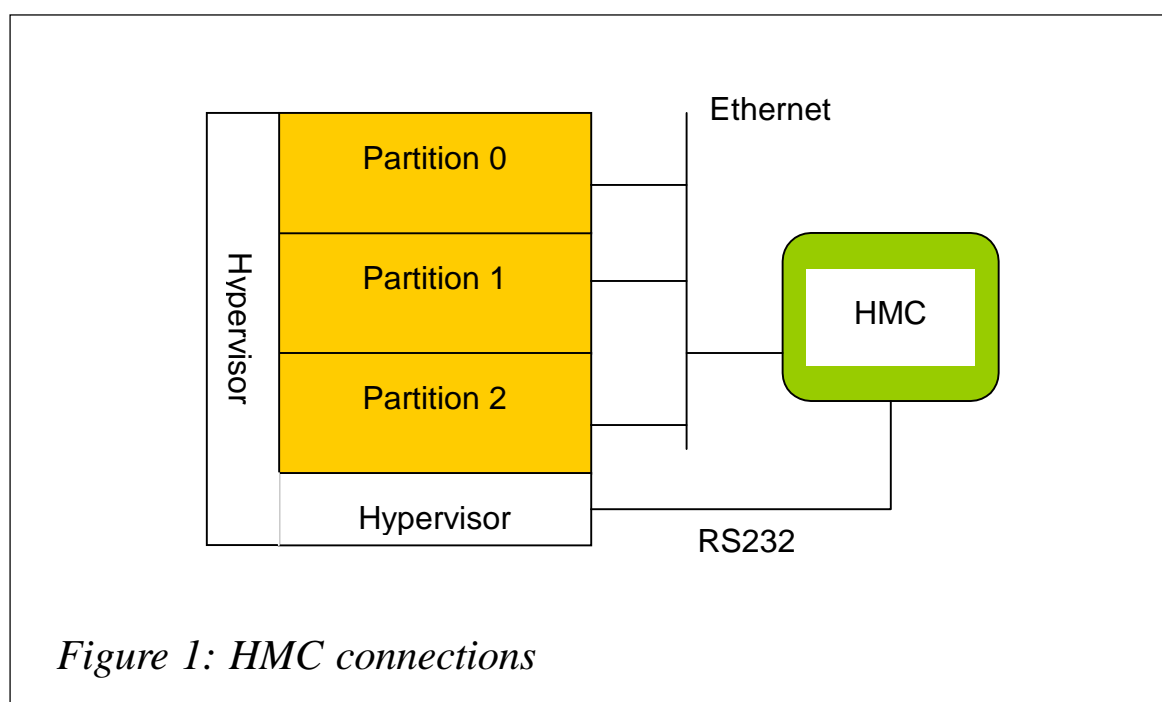


Figure 1: HMC connections

Figure 1). A hypervisor is firmware that manages the whole server and the partition boundaries and on which DLPAR is built.

The HMC presents the server administrator with a GUI, which reduces the complexity of DLPAR operations to a sequence of mouse clicks.

The HMC GUI is simply a wrapper around a number of DLPAR-specific Linux commands pre-installed on the HMC. The DLPAR commands can be found in `/opt/hsc/bin` on the HMC. As of Revision 3v22 of the HMC, it is possible to remotely execute these commands via a remote shell. Alternatively, with a little effort, it is possible to develop scripts to perform common DLPAR

operations that may be executed from the command line or automatically with a cron.

Given the power and nature of the HMC DLPAR operations and the potential chaos that may be brought by an inappropriate DLPAR configuration, access to the HMC is generally limited to a privileged few. This restricted access may create a number of difficulties in day-to-day operations because a number of DLPAR operations occur frequently and involve little or no risk to the system security or integrity – for example moving a CD drive from one partition to another in order to install some software. In such cases the intervention of the system administrator, though rarely justified, is mandatory because only they know the HMC system password. Ideally in such circumstances, the administrator (or even end user) of the partition requiring the CD would use a remote shell (**rsh** or **rexec**) to run the appropriate commands on the HMC in order to perform the transfer.

A potential problem with this approach is that the standard **rsh/rxec** commands move data and passwords in clear-text. In many production environments this is not appropriate. Both the HMC and AIX support a secure version of remote shell known simply as secure shell, or ssh. Ssh ciphers data exchanged between the communicating parties, making it extremely difficult to eavesdrop on any conversation. Using ssh is no more complicated than **rsh/rxec**. The remainder of this article explains how to install, configure, and use ssh between AIX and an HMC.

SECURE SHELL – SSH

SSH (secure shell) is a program that secures network communication over insecure networks.

SSH was designed to replace the **rx** commands (**rsh**, **rcp**, and **rlogin**) that came from BSD Unix. It now also secures other remote applications such as telnet, ftp, and X11. Ssh provides authentication of the communicating parties and encryption of the data exchanges. This permits you to securely:

- Log in to remote computers.

- Execute commands on a remote computer.
- Copy files to and from a remote computer.
- Forward X11 sessions.

Arbitrary TCP/IP ports can be forwarded over the secure channel, enabling secure connection, for example, to an e-mail service.

CONFIGURING SSH ON THE HARDWARE MANAGEMENT CONSOLE

As the HMC runs on a Red Hat Linux PC, the **ssh** application comes from an open source project and is known as Open Secure Shell or OpenSSH.

Use the menu *HMC Maintenance/System Configuration/Remote Execution Option* and check the option *Enable remote command execution using the ssh facility* – see Figure 2.



Figure 2: Remote execution option

INSTALLING THE OPENSHELL CLIENT ON AIX5L 5.2

The OpenSSH software is shipped on the AIX 5.2 Bonus Pack, packaged as an installp package. At the time of writing the latest version is openssh-3.6.1p2. OpenSSH can be found on the AIX Toolbox for Linux Applications CD.

Before installing the OpenSSH packages, you must install the Open Secure Sockets Layer (OpenSSL). OpenSSL includes the encryption library required by OpenSSH. OpenSSL can be found as an RPM package on the AIX Toolbox for Linux Applications CD.

The OpenSSL and OpenSSH filesets can be installed with `smit install_software` (even though OpenSSH is packaged as an RPM). Alternatively you can use the **geninstall** command:

```
# geninstall -d/dev/cd0 R: openssl -0.9.6g
# geninstall -d/dev/cd0 I: openssl.base
```

The installation of OpenSSH adds an entry to the `/etc/rc.d/rc2.d` directory and to **inittab**, which will automatically start the `sshd` daemon when the server (or partition) boots. To prevent the daemon from starting automatically you should delete the files `Ksshd` and `Ssssd` in the directory `/etc/rc.d/rc2.d`.

The OpenSSH daemon, `sshd`, is located in the directory `/etc/ssh` along with the `ssh` client configuration files. The `sshd` runs under the system resource controller, SRC; thus, starting and stopping the daemon and getting its status can be performed using the standard **src** commands:

```
startsrc | stopsrc | lssrc -s sshd
```

OpenSSH adds the following commands:

- **scp** (secure copy) – a remote file copy program similar to `rcp`.
- **sftp** (secure ftp) – an ftp client program that runs over SSH1 and SSH2 protocols.
- **sftp-server** (secure ftp server subsystem) – similar to `ftpd`. This service is started automatically by the `sshd` daemon.
- **ssh** (secure shell) – a secure version of the `rlogin` and `rsh` clients.
- **ssh-add** – adds private keys to `ssh-agent` (see next command).

- **ssh-agent** – a software component that stores private keys.
- **ssh-keygen** – a key generation tool for asymmetric keys.
- **ssh-keyscan** – collects the public keys held by remote hosts.
- **ssh-keysign** – allows a host to act as a certificate authority and to sign a public key.
- **sshd** (the secure shell daemon) – handles incoming SSH1/2 connection requests.

TESTING THE SSH CONNECTION

To verify the installation procedure we will remotely execute the command **date** on the HMC from AIX. For this we will use the account *hscroot* on the HMC. Log on to AIX and execute either of the following two commands:

- `ssh -l hscroot hmc date`
- `ssh hscroot@hmc date.`

These two commands show the different syntaxes for specifying the remote host and remote user with **ssh**. *hscroot* is the username and *hmc* is the IP name of the HMC. Either of these commands will produce something like this:

```
The authenticity of host 'hmc (174.26.13.246)' can't be established. RSA
key fingerprint is a7:ef:33:51:aa:a2:cc:57:85:69:81:00:12:76:22:00.
Are you sure you want to continue connecting (yes/no)?
```

Reply *yes* and you will see the following message:

```
Warning: Permanently added 'hmc, 174.26.13.246' (RSA) to the list of
known hosts.
hscroot@hmc's password:
```

After providing the password for *hscroot* you will see the result of the command **date**, giving the current time on the HMC.

CONFIGURING SSH

The above test highlights two important points. Firstly we used

the user hscroot on the HMC. This is inappropriate in most cases; we should create one or more specific accounts at least on the HMC. Secondly we had to provide manually the hscroot password to execute the remote command. This makes it difficult to use **ssh** commands from shell scripts and the cron. To avoid being prompted for a password we need to generate a public key on AIX and copy it to the HMC.

Create a new user account using the HMC management menus with the roles *System Administrator* and *Advanced Operator*. This is the user that will be used for the remote execution of DLPAR operations – for the examples below I will call the account *dlpar*. It may be appropriate to create a new HMC user for each AIX partition on the server, each account having a specific set of commands available. This gives close control over who can do what.

Before creating a public key on AIX you need to create a directory *\$HOME/.ssh*; this is the directory where both public and private keys will be held. You use the following command to create your key pair. Should you prefer the DSA algorithm over that of RSA, replace the parameter **rsa** by **dsa** – in most cases the choice of one algorithm over the other does not have any significant effect.

```
ssh-keygen -t rsa
```

The command produces a number of informational messages. Choose the default locations for the key files. When the command has finished you will find your public and private keys in the files *id_rsa* (or *id_dsa*) and *id_rsa.pub* (or *id_dsa.pub*) respectively in the directory *\$HOME/.ssh*.

The above operations have to be repeated for each AIX account that will use **ssh** to connect to the HMC.

Now we have our keys pair, we have to make them visible to the HMC. First of all, log in to the HMC as the user that you created for performing the DLPAR operations and create the directory *~dlpar/.ssh* (you can do this by **ssh** if you wish). From AIX copy the public key to the HMC using secure copy **scp**:

```
cd ~/.ssh
```

```
scp id_rsa.pub dlpar@hmc: .ssh/  
scp id_dsa.pub dlpar@hmc: .ssh/
```

You are then prompted for the password for the account dlpar, and the file is copied to the HMC called hmc.

Then on the HMC append the public key to the authorized key list:

```
cd ~/.ssh  
cat id_rsa.pub >> authorized_keys  
cat id_dsa.pub >> authorized_keys2
```

You can then delete the public key file:

```
rm id_?sa.pub
```

Again, this operation should be repeated for each partition that will use **ssh** to access the HMC. Note that you should not copy the public key file directly into the authorized key list because this will delete any keys previously installed. Obviously you could use the **ssh** command to perform the concatenation and removal operations.

Now everything is in place and when you execute the **date** command (**ssh -l dlpar hmc date** or **ssh dlpar@hmc date**) no password is required and the date on the HMC is returned.

REMOVING KEYS

Removing a key on the HMC will not prevent the key owner from using **ssh** commands, but simply make him/her provide his/her password for each command. To do this edit the file `~/.ssh/authorized_keys` or `~/.ssh/authorized_keys2` and delete the line containing the key concerned.

CLOSING REMARKS

Not so long ago security products were expensive, often complicated, custom-built solutions. Today they are frequently bundled free with the operating system, simple to use, and effective. Many quality solutions, such as **ssh**, TLS (Transport Level Security), and PGP (Pretty Good Privacy), are available as open source. Tools such as **ssh** play an important part in today's networked environment.

The use of **ssh** does nothing more than secure the communication channels. It does not serialize requests. As such you must take steps to ensure that two users do not perform conflicting DLPAR operations concurrently – this is true independent of whether **ssh** is used or not.

DLPAR operations that move I/O slots from one partition to another will work only if the devices connected to the slot are not mounted on the source partition. In such cases a script is often necessary to perform the unmount operation. This script would typically run on the HMC and remotely execute the AIX commands in the partition concerned. Clearly **ssh** should be used for this operation too. In this case, it is necessary to generate public keys for the account `dlpar` on the HMC and add them to the trusted keys list of a DLPAR account on each of the partitions.

HMC COMMANDS

When using **ssh** it is possible to execute any command or script that is in the `PATH` of the HMC user account to which you are connected. Generally these are the standard GNU/Linux commands along with the DLPAR commands in `/opt/hsc/bin`. Here is a brief resumé of what you will find in this directory.

Commands for configuration and administration of the HMC:

- **lshmc** – display the HMC configuration: network, VPD, remote command, etc.
- **chhmc** – change the HMC configuration.
- **hmcshutdown** – shut down the HMC GNU/Linux operating system.
- **lshmcusr** – show an HMC user's profile.
- **chhmcusr** – modify an HMC user's profile.
- **mkhmcusr** – create a user account on the HMC.
- **rmhmcusr** – delete a user account on the HMC.
- **bkprofdata** – back up HMC profile data to a single file on the

HMC (this can then be copied remotely for safekeeping).

- **rstprofdata** – restore the HMC profile from a file.
- **lssvcevents** – list the HMC events gathered from the managed partitions.

LPAR management commands:

- **lshwinfo** – show hardware information: temperature, etc.
- **lscuod** – show information related to Capacity Upgrade on Demand (CUoD).
- **chcuod** – change CUoD attributes.
- **rmsplock** – remove a lock set on the service processor of a managed system.

Partition management and configuration commands:

- **lssyscfg** – list system and partition profiles and hardware resource configurations.
- **chsyscfg** – change partition profile.
- **mksyscfg** – create a new partition and/or partition profile.
- **rmsyscfg** – delete a managed system, system profile, partition, or partition profile.
- **chsysstate** – change system state. Used to power on and off or reset partitions. To start a partition that is in the ready state (after shutdown –F or reset) use:

```
chsysstate -r lpar -m <machine> -n <partition> -o on
```

- **lshwres** – list the hardware resources allocated to a system or a partition.
- **chhwres** – dynamically change (without reboot) the hardware resource configuration, DLPAR.
- **rsthwres** – restore a hardware resource configuration following a failure of the above command.

- **mkvtterm** – open a virtual terminal session to a given partition – creates an xterm-like window with a login shell.
- **rmvtterm** – close a virtual terminal session opened with the above command.

REFERENCES

AIX documentation:

- *HMC Installation and Operations Guide*.
- *AIX 5L v5.2 Security Guide SC23-4860*.

Redbook: *Effective System Management Using the IBM HMC for pSeries SG24-7038*.

Web links:

- http://www-1.ibm.com/servers/esdd/articles/openssh_updated.html
- <http://oss.software.ibm.com/developerworks/projects/opensshi>
- <http://www.openssh.org>
- <http://www.openssl.org>.

Jez Wain
System Architect
Groupe Bull (France)

© Xephon 2004

LPAR memory management

With the pSeries IBM offers a new technology to AIX, which is called Logical PARTitioning (LPAR). Although this technology was introduced almost 20 years ago for MVS, it's completely new to AIX.

The following models of the pSeries offer LPAR support:

- pSeries 630 Models 6C4 and 6E4
- pSeries 650 Model 6M2
- pSeries 655 Model 661
- pSeries 670 Model 671
- pSeries 690 Model 681.

Logical partitioning means that a multiprocessor computer is divided into multiple independent operating systems using software and firmware to control resources like CPUs, memory, and I/O slots.

Although the above-mentioned computers can run only one AIX system (called full system partition mode), they are more likely to build several logical partitions. To achieve this a separate Intel-based desktop workstation, running Red Hat Linux, is available. The workstation is called Hardware Management Console (HMC) and is the control point for multiple LPAR systems.

All definitions and changes to an LPAR-systems are managed via HMC using Java-based applications. The usage of these Java-based applications is very intuitive, but you should prepare thoroughly before you begin with the definition of logical partitions. This is especially important for the memory definitions. At first I will cover some major issues about the memory usage in an LPAR system and then explain some details of other resources like processors and I/O slots.

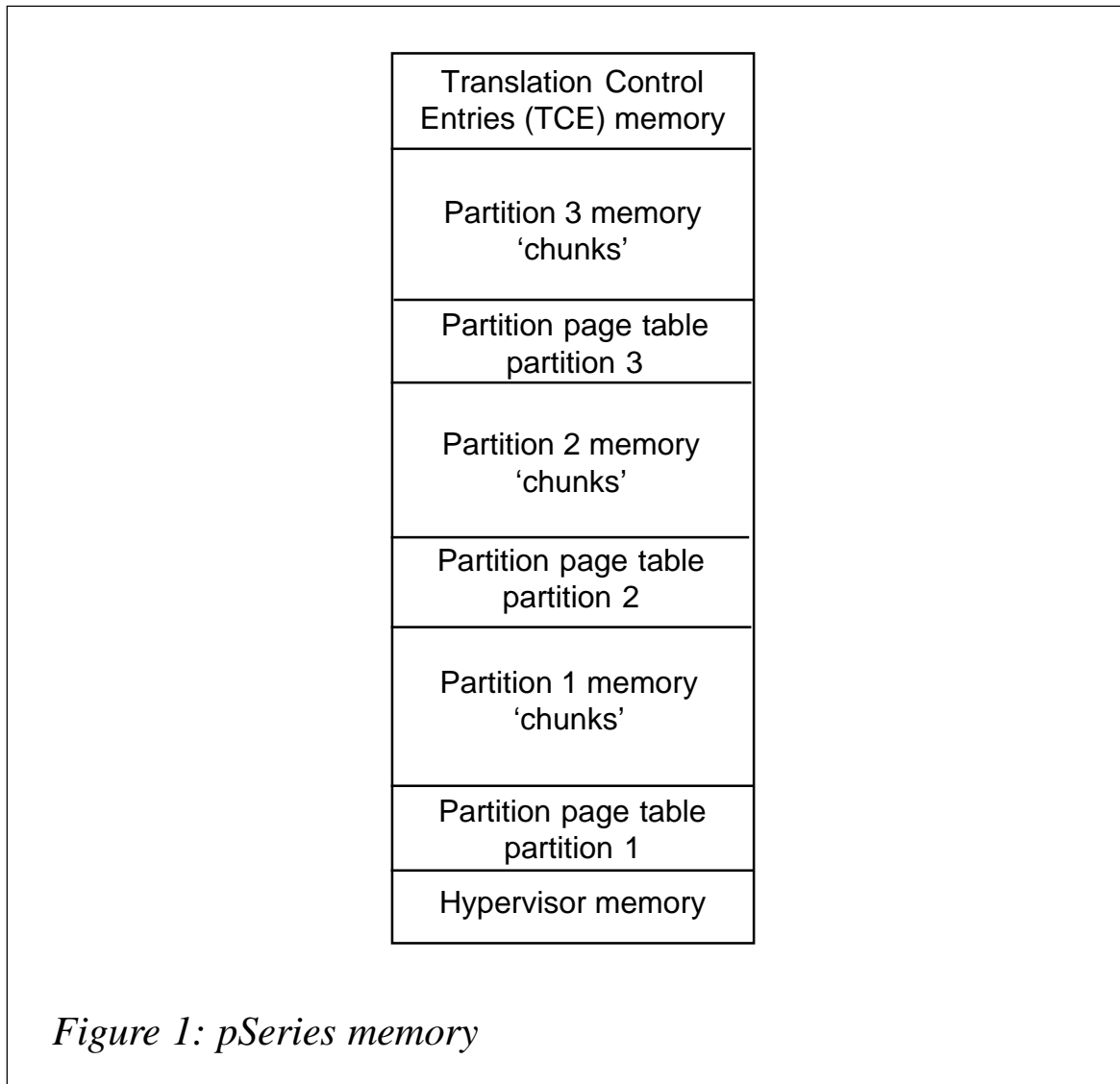
MEMORY RESOURCES

To understand the memory management in an LPAR system, first we should take a look at the memory usage of a pSeries system when powered on to support partitions. As you can imagine there will be a memory overhead when dividing up the memory between partitions.

Figure 1 shows how the memory of a pSeries system is divided:

Hypervisor memory

The hypervisor is simply a piece of software controlling the logical partitions. The hypervisor allocates the first 256MB of a



pSeries system independently of the number of partitions or the overall size of the system. The hypervisor memory is always located at address 0 in the physical memory. No hypervisor memory is used if the system runs in full system partition mode.

Translation Control Entries memory

This part of the memory translates the I/O addresses to the

system memory. This memory is also independent of the number of partitions, but it depends on the number of I/O drawers in the system. The first four I/O drawers require a total of 256MB of memory. If the systems has five to eight I/O drawers another 256MB of memory is required. This memory is allocated for the total number of I/O slots in a system regardless of whether they are used or not. It makes no difference whether an I/O slot contains an adapter or not. TCE memory is always allocated (in both full system partition mode and logical partition mode) and, as you can see in Figure 1, always at the top of the physical memory.

In other AIX systems, TCE memory is part of the regular memory that AIX allocates at boot time. Usually an LPAR system may contain many AIX operating system environments. This TCE memory is allocated separately from the partition memory.

Partition page tables

Page tables map the physical memory pages to virtual memory pages. Each partition requires $1/64^{\text{th}}$ of its memory size, rounded to a power of 2, for each page table in memory.

For each partition, a 256MB segment of page table space is allocated. How much memory a partition actually allocates in that segment depends on the maximum partition memory. The dependency between partition memory and page table space is shown in Figure 2.

Looking closely at Figure 2, partition memory is always assigned in 256MB segments. The minimum amount of memory you can assign for one partition is 1GB for AIX 5.1 and pre-10/2002 firmware. For AIX 5.2, the minimum amount of memory you can allocate is 256MB. As mentioned above, partition page table space is always allocated in 256MB segments. But if you have partitions that require less than 256MB of page table space, these partitions can share one 256MB segment. For example if you have two partitions each with 8GB, they share one 256MB page table segment because one 8GB partition requires 128MB

<i>Maximum partition memory</i>	<i>Page table space</i>
256MB	4MB
512MB	8MB
768MB	16MB
1GB	16MB
1.25 – 2GB	32MB
2.25 – 4GB	64MB
4.25 – 8GB	128MB
8.25 – 16GB	256MB
16.25 – 32GB	512MB
32.25 – 64GB	1GB
64.25 – 128GB	2GB
128.25 – 256GB	4GB
> 256.25GB	8GB

Figure 2: Partition memory and page table space

of page table memory. This leaves another 128MB of memory in the page table free. So two 8GB partitions need 256MB of page table space.

Before we can go into the details of how to calculate the memory resources we have to talk about some other issues: minimum/desired/maximum, small real mode memory, and 'large real mode memory'.

Minimum/desired/maximum

When you define the memory allocation for a partition you must enter the following three different values:

- Minimum amount of memory means that the system must be able to allocate the minimum amount, otherwise the partition will not start.
- Desired amount of memory means that the system will attempt to allocate memory up to the desired amount of memory, if it's available, when the partition starts. The increment is 256MB.

- Maximum amount of memory means that this amount of memory can be used for dynamic allocations. This amount is also used when calculating the total amount of memory needed for the page tables to support this partition.

Large real mode memory

Real mode memory is a contiguous area in physical memory that AIX needs for system operations. The location of this memory is known to the operating system. Each partition has one real mode memory address region and it starts at that region's 'logical' memory address, 0.

A partition allocates a contiguous amount of memory when it starts. If the real mode memory region is 16GB, then an area of 16GB of contiguous memory must be found for the partition to start. To make things even worse, this memory must fall on a memory boundary that is a multiple of the amount. Here are two examples. For 1GB of real memory the memory boundaries are 1, 2, 3GB and so forth. For 16GB of real memory, the boundaries are 16, 32, 48GB, and so forth. As already mentioned, the hypervisor locates its memory always at the lowest memory address, so the 0GB boundary can never be used. This means that insufficient planning of memory allocation may lead to situations where partitions fail to start, although there is plenty of memory available, because there is no contiguous memory large enough and at the right memory boundary for the real mode memory.

Small real mode memory

For AIX 5.1 small real mode memory is not available. So for AIX 5.1 there are only two possible values of real mode memory sizes:

- Partitions \leq 16GB – real mode size is 1GB (that is also the minimum amount of memory that must be allocated).
- Partitions $>$ 16GB – real mode size is 16GB.

Be aware that if you set a small real mode address region for AIX 5.1 partitions, AIX will fail to boot. The operator panel on the HMC will show 'Invalid LPAR Partition'. In the console window you will see 'Fatal error: Invalid LPAR Partition'.

Small real mode memory is available for AIX 5.2 and Linux. The HMC must run minimum software release 10/2002. Small real mode should always be used for AIX 5.2 and Linux.

The minimum for small real mode memory is 256MB.

Now let's have a practical look at what all this means for calculating the memory overhead.

Example 1

We have a system with the following configuration:

- 64GB of total system memory.
- Four I/O drawers.
- Three partitions with these memory sizes: 16GB, 8GB, and 4GB.

What will the total memory overhead be?

Hypervisor: 256MB

TCE: 256MB

Partition 1: 256MB

Partition 2: 128MB

Partition 3: 64MB

Partition 2 and partition 3 allocate their page table memory in one 256 memory segment ($128+64=192$), so the total overhead will be 1GB.

Example 2

We have a system with the following configuration:

- 64GB of total system memory.
- Six I/O drawers.
- Four partitions with these memory sizes: 16GB, 8GB, 4GB, and 2GB.

What will the total memory overhead be?

Hypervisor: 256MB

TCE: 512MB (because more than 4 I/O drawers are used)

Partition 1: 256MB

Partition 2: 128MB

Partition 3: 64MB

Partition 4: 32MB

Partition 2, partition 3, and partition 4 allocate their page table memory in one 256 memory segment (128+64+32=224), so the total overhead will be 1.25GB.

Total memory (GB)	Approx. memory overhead (GB)	Approx. usable partition memory (GB)	Max. partitions <= 16 GB prior to 10/2002 firmware	Max. partitions > 16 GB post 10/2002 firmware Large	Max. partitions all sizes post 10/2002 firmware Small
2	.75 – 1 GB	1 - 1.25 GB	0/0	5/0	5
4	.75 – 1 GB	3 - 3.25 GB	2/0	13/0	13
8	.75 – 1 GB	7 - 7.25 GB	6/0	16/0	16
16	.75 – 1 GB	15 – 15.25 GB	14/0	16/0	16
24	1 – 1.25 GB	22.75 – 23 GB	16/0	16/0	16
32	1 – 1.5 GB	30.5 – 31 GB	16/0	16/0	16
48	1.5 – 2 GB	46 – 46.5 GB	16/1	16/1	16
64	1.5 – 2.25 GB	61.75 – 62.5 GB	16/2	16/2	16
96	2 – 3.5 GB	92.75 – 94 GB	16/4	16/4	16
128	2.5 – 4 GB	124 – 125.5 GB	16/6	16/6	16
192	3.5 – 5.75 GB	186.25 – 188.5 GB	16/10	16/10	16
256	4.5 – 7.5 GB	248 – 251.5 GB	16/14	16/14	16

Figure 3: Memory overhead

PARTITIONS BETWEEN 2GB AND 256GB

Figure 3 shows the memory overhead for various sizes of system memory. It also shows what's left over as 'usable' partition memory. It also shows the maximum of partitions for different types of partition and firmware releases.

The numbers with a slash (/) show the maximum number of small partitions (<= 16GB) that can be started followed by the number of those that can be 16.25GB or larger. Note that for large real memory region partitions, you must have at least 48GB of physical memory to start one partition that requires more than 16GB of memory.

Total memory (GB)	Approx. memory overhead (GB)	Approx. Usable partition Memory (GB)	Maximum partition size AIX 5.1 <=16GB/>16GB	Maximum partition size AIX 5.2 or Linux
320	5.5 – 9.25 GB	310.75 – 314.5 GB	32/18	32
384	6.5 – 11 GB	373 – 377.5 GB	32/22	32
448	7.5 – 12.75 GB	435.25 – 440.5 GB	32/26	32
512	8.5 – 14.5 GB	497.5 – 503.5 GB	32/30	32

Figure 4: Maximum number of partitions

PARTITIONS LARGER THAN 256GB

A complete chart for the maximum partitions and the overhead for various operating system versions can be found in the *IBM HMC for pSeries Installation and Operating Guide*.

Figure 4 shows the maximum number of partitions for a pSeries 690 with post 05/2003 firmware. The HMC must run Release 3.1.

PARTITION STARTING ORDER

Another important issue is the order in which partitions with different sizes should be started. Before defining the memory sizes you should also check the memory allocation required by the application running in a partition.

For small real mode memory partitions, just analyse the partition profile to make sure your partitions will all start with their desired, or at least their minimum, memory. The maximum memory value should not be unreasonably high so that memory is not wasted on page table space.

Large real mode memory partitions:

- First start all the large memory partitions with memories larger than 16.25GB. Start the smallest of these large memory partitions first. Work your way up to the largest partition.
- Start your small memory partitions with memory equal or less than 16GB next. Start the smallest of these small memory partitions first. Work your way up to the largest partition.
- If you still have problems starting all the partitions, try to reduce the memory requirements of one or more partitions. Remember that partitions larger than 16GB memory always require 16GB of contiguous memory. If the memory is equal to or less than 16GB, only 1GB of contiguous memory is required.

If you have small and large real mode memory partitions on the same system, start the large real mode memory partitions first.

Here is an example of the partition start order for large real mode memory:

Assumption:

- p690 with 64GB memory.
- Three large real mode memory partitions (AIX 5.1):
 - Partition1: 30GB
 - Partition2: 24GB
 - Partition3: 5GB.
- Five I/O drawers.

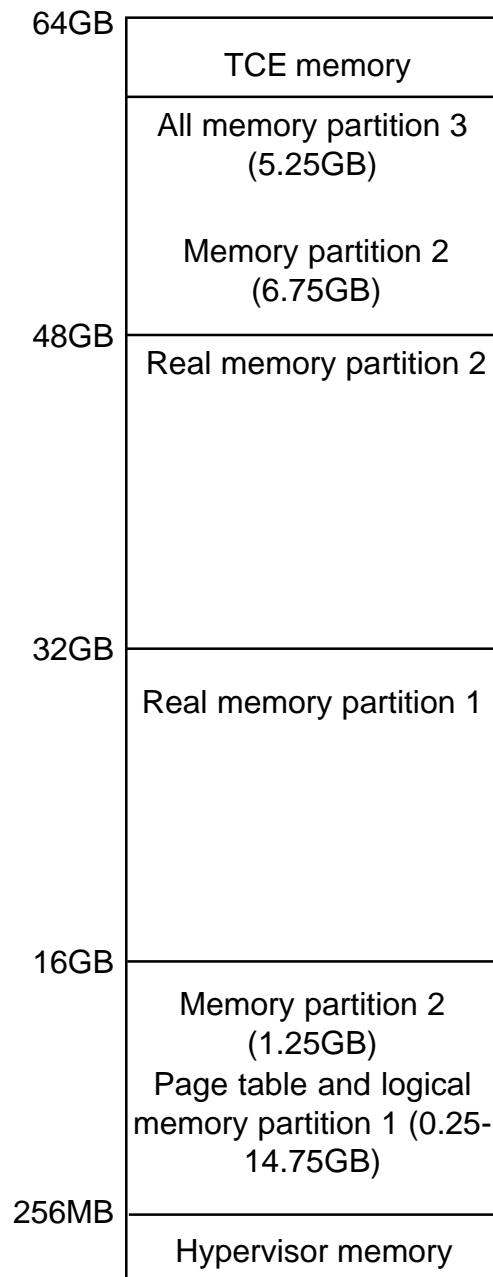


Figure 4 : Starting order

- Memory overhead:
 - Hypervisor: 256MB
 - TCE memory: 512MB

- Page table memory: 512MB + 512MB + 256MB.

Figure 4 shows why the starting order of the partitions with large real memory is so important. As already mentioned, real mode memory must be at a physical memory boundary that is a multiple of the amount of real mode memory and the real mode memory must be allocated in a contiguous area.

Now let's have a look at partition starting order for two examples. The first example will be successful for all partitions, the second example will start only two partitions.

Example 1 – starting order is partition1, partition2, and partition3:

- Partition1 allocates 16GB real mode memory at the 16GB memory boundary. Then .5GB memory is allocated for the page table. Then the partitions allocate 256MB free memory segments starting from the bottom up to fill the rest of the memory requirements.
- Partition2 finds the next available 16MB area of contiguous memory at the 32GB memory boundary. Then .5GB memory is allocated for the page table. Then it allocates 256MB memory segments starting from the bottom up to fill the rest of the memory requirements.
- Partition3 requires only 1GB of real mode memory. This will be no problem to find because memory boundaries will be at 1GB. First .5GB memory is allocated for the page table, then it allocates 256MB memory segments starting from the bottom up to fill the rest of the memory requirements

Example 2 – starting order is partition1, partition3, and partition2:

- Partition1 allocates 16GB real mode memory at the 16GB memory boundary. Then .5GB memory is allocated for the page table. Then the partition allocates 256MB free memory segments starting from the bottom up to fill the rest of the memory requirements.
- Partition3 allocates the next available contiguous 1GB memory, that is bound to 1GB boundary. That's at address

15GB – 16GB. Then 128MB memory is allocated for the page table. That's at address 14.75GB – 15GB. Because another 4GB are still required, the rest of the memory is allocated from 32 – 36GB.

- Partition2 needs 16GB of contiguous memory at a 16GB (or multiple) boundary, but no memory that satisfies all the requirements is found. Partition2 will not start.

As you can see the starting order of the partitions may cause a lot of trouble if they are started in the wrong order. It also shows the importance of resource planning.

I/O RESOURCE

A partition requires at least one boot device and one network adapter in an allocated I/O slot. Usually other I/O adapters are also required for storage and network access. I/O slots are located in a pSeries machine and/or additional I/O drawer(s).

When defining I/O resources to a partition there are two options – required and desired.

Slots that are defined as required *must* be available when the partition is started. If a required slot is not available the start of the partition will fail. Required slots cannot be dynamically removed or moved to other partitions.

Slots that are defined as desired are assigned to a partition if they are available at starting time. Partition start will be successful even if the desired slot is not available. Desired slots can be dynamically removed from the partition and can be assigned to other partitions.

PROCESSOR REQUIREMENTS

A partition requires one processor (minimum), all of the processors on the system, or any number between.

When defining processor resources to a partition, we also have to deal with the minimum/desired/maximum value.

Minimum setting

The minimum value defines how many processors are required for the partition to start. If the minimum requirements are not fulfilled the partition will fail to start.

Desired setting

The desired value defines how many processors a partition can use if they are available. When a partition is started, the system allocates the number of available processors up to the desired number.

Maximum setting

The maximum value defines the largest number of processors that can be assigned to this partition. If processors are dynamically added to a running partition this setting defines the maximum number of processors for this specific partition.

STARTING ORDER

When defining several partitions you must always keep the starting order in mind. The minimum and desired settings are the ones that must be observed. Here is an example of the partition start order:

Assumption:

- pSeries with 16 processors.
- Three partitions with the following values (minimum/desired/maximum):
 - Partition1 6/8/10 processors.
 - Partition2 2/4/16 processors.
 - Partition3 4/10/16 processors.

Example 1 – starting order is partition1, partition2, and partition3:

- Partition1 starts and allocates eight processors

- Partition2 starts and allocates four processors
- Partition3 starts and allocates four processors. More processors cannot be allocated because all 16 processors are used, but the minimum requirement of four processors is fulfilled.

Example 2 – starting order is partition1, partition3, and partition2:

- Partition1 starts and allocates eight processors.
- Partition3 starts and allocates eight processors.
- Partition2 cannot start because all 16 processors are already allocated.

FINAL HINT

Document, document, document!

A very important issue is documentation. Keep track of all definitions and changes you make to any partitions. For the configuration planning you will find a planning worksheet in the IBM book *Planning for Partitioned-System Operations SA38-0626-00*.

Robert Schuster
System Administrator (Germany)

© Xephon 2004

Using HMC command line interface to manage Logical Partitioning Facility (LPAR)

INTRODUCTION

My previous article ('Logical Partitioning Facility (LPAR) planning and implementation', *AIX Update*, issue 99, January 2004) discussed basic planning and implementation of logical

partitioning in the environment of IBM eServers pSeries computers running AIX operating system. I didn't discuss the actual set-up and management of LPARS, which is based on utilization of IBM Hardware Management Console for pSeries (HMC).

The HMC is a dedicated PC computer, running the Red Hat Linux operating system. It has a Web-based GUI, which is fairly simple and logical to operate. This article discusses the command line interface of HMC that can be used in order to automate LPAR-related operations and to apply consistent settings to several servers that have to be configured in a similar way.

I will use the p650 server, the setting up of which was discussed in the previous article, to demonstrate various commands.

SYNTAX AND COMMON HMC COMMAND FLAGS

In this section I will cover the syntax and common flags used by different HMC commands. All examples are performed on HMC running Version 3.2.4 of the operating software.

First let's execute a single command such as **lssyscfg**.

Usage:

```
lssyscfg -r "resource type" -n "system object name" | -all  
[-m "managed system"][-p "source partition name"][-F<"format"> | -z ]  
-help - print this message.
```

...

The following conventions are used:

- Parameters not enclosed in brackets must be entered literally.
- Items enclosed in angle brackets, such as <"format">, represent variables that should be replaced by a relevant value.
- Items enclosed in square brackets, such as [-help], are optional.
- Items enclosed in curly brackets, such as {}, are required.

- A vertical bar indicates that only one value has to be chosen.
- The flag and the parameter value should be separated by blank(s).
- If the parameter or value contains blanks or special characters expanded by the shell, they should be included in quotes.

The **-m** flag specifies the managed system on which the HMC operation is to be performed. This flag can be used with one of the following types of argument:

- The managed system name.
- The machine type, model, and serial number of the managed system in the following format: MT-MDL*S/N, where MT is the machine type, MDL is the model number, and S/N is the serial number of the managed machine. For instance F7038-6M2*632C33A.

The **-r** flag specifies the resource of the managed system on which you wish to perform the operation. The **-r** flag can be specified with the following arguments:

- cpu – CPU
- mem – memory
- slot – adapter slot
- led – LED
- alpar – affinity partition
- lpar – logical partition
- prof – partition profile
- sysprof – system profile
- sys – manage system
- frame – frame
- ALL – all possible types of resources.

The **-n** flag specifies the name of the specific object on which you wish to perform the command. This flag is mutually exclusive with the **-all** flag that specifies selection of all objects.

The **-o** flag specifies the operation that we want to perform on the resource. For instance one of the arguments of the **chsysstate** command must be specified as **-o <"operation">**, with the following possible values for the operation field:

- **on** – to perform a power on or activation of a partition.
- **off** – to perform a power off of a managed system or hard reset of a partition.
- **osreset** – to reset the operating system on the partition.
- **osshutdown** – to shut down the operating system on the partition.
- **reset** – to perform a soft reset of a partition.
- **rebuild** – to rebuild a managed system.
- **recover** – to recover partition data.

The **-p** flag specifies the name of the specific partition on which you wish to perform the command. For a DLPAR move operation the **-p** flag specifies the source partition of the operation.

The **-f** flag specifies the name of a file. This flag is used by the **bkprofdata** and **rstprofdata** commands in order to specify the name of the profile data file to be used for output or input for back-up or restore operations. When used with the **mksyscfg** or **chsyscfg** commands, it specifies the configuration data file to be used as input of the operation.

The **-F** flag specifies which formatted fields are to be used when displaying output from commands starting with the **ls** prefix. This command enables the selective and ordered display of information retrieval commands.

The **-help** is used to display the syntax and sample use of the command.

HMC MANAGEMENT COMMANDS

Ishmc

The **Ishmc** command is used to list the HMC network configuration, the remote command configuration, and VPD data:

```
Ishmc [ -n ] [ -v ] [ -r ] [ -F <format> ] [ -help ]
```

where:

- **-n** – displays network information.
- **-v** – displays the console's VPD information.
- **-r** – displays remote command execution settings.
- **-F** – if specified, a delimiter-separated list of property names to be queried. Valid values are **hostname**, **domain**, **nameserver**, **domainsuffix**, **gateway**, **ipaddr**, **networkmask**, and **ssh**.
- **-help** – prints a help message.

The following are some examples of the usage of this command:

```
[hmcusr@hmcproj hmcusr]$ Ishmc -n
Network Configuration:
Host Name:                hmcproj
TCP/IP Interface 0 Address: 192.168.2.53
TCP/IP Interface 0 Network Mask: 255.255.255.0
Default Gateway:         192.168.2.254
Domain Name:              mycompany.com
DNS Server Search Order:  192.168.2.6
Domain Suffix Search Order:
```

```
[hmcusr@hmcproj hmcusr]$ Ishmc -v
Vital Product Data Information:
*FC ??????????
*VC 20.0
*N2 Fri Dec 05 21:52:53 IST 2003
*FC ??????????
*DS pSeries Hardware Management Console
*TM eserver xSeries 335 -[7343CR2]-
*SE 651A99A
*MN IBM
*PN Unknow
*SZ 1055932416
```

```
*OS Linux 2.4.20-19.7
*NA 192.168.2.53
*FC ?????????
*DS Platform Firmware
*RM R3V2.4
```

```
[hmcusr@hmcproj hmcusr]$ lshmc -r
Remote Command Execution Configuration:
Remote command execution using the ssh facility: enabled
[hmcusr@hmcproj hmcusr]$ lshmc -n -Fhostname:ipaddr:
hmcproj:192.168.2.53:
```

chhmc

The **chhmc** command is used to change the HMC network configuration or to enable and disable remote command execution:

```
chhmc -c [network | ssh] -s [enable | disable | add | modify | remove]
      [-i eth0 | eth1 [-a ip-address] [-nm network-mask]]
      [-d network-domain-name] [-h host-name]
      [-g gateway] [-ns DNS-Server] [-ds Domain-suffix]
      [-help]
```

where:

- **-c** – the type of configuration to modify. Valid values are **ssh** and **network**.
- **-s** – the new state value of the configuration. When the configuration type is **ssh**, the valid values are **enable** and **disable**. When the configuration type is **network**, the valid values are **add**, **modify**, and **remove**. **Add** and **remove** are valid only when specifying **-ns** or **-ds**.
- **-i** – the interface to configure. Valid values are **eth0** and **eth1**. This parameter can be used only with **-s modify**.
- **-a** – the new network IP address. This parameter can be used only with the **-i** parameter.
- **-nm** – the new network mask. This parameter can be used only with the **-i** parameter.
- **-d** – the new network domain name. This parameter can be used only with **-s modify**.

- **-h** – the new host name. This parameter can be used only with **-s modify**.
- **-g** – the new gateway address. This parameter can be used only with **-s modify**.
- **-ns** – the DNS server to add or remove. This parameter can be used only with **-s add** or **-s remove**.
- **-ds** – the domain suffix to add or remove. This parameter can be used only with **-s add** or **-s remove**.
- **-help** – prints a help message.

The following are examples of the usage of this command.

To add and remove another domain suffix search other than amodocs.com:

```
[hmcusr@hmcproj hmcusr]$ lshmc -n
Network Configuration:
Host Name:                hmcproj
TCP/IP Interface 0 Address: 192.168.2.53
TCP/IP Interface 0 Network Mask: 255.255.255.0
Default Gateway:         192.168.2.254
Domain Name:             mycompany.com
DNS Server Search Order: 192.168.2.6
Domain Suffix Search Order: mycompany.corp
[hmcusr@hmcproj hmcusr]$ chhmc -c network -s remove -ds mycompany.corp
[hmcusr@hmcproj hmcusr]$ lshmc -n
Network Configuration:
Host Name:                hmcproj
TCP/IP Interface 0 Address: 192.168.2.53
TCP/IP Interface 0 Network Mask: 255.255.255.0
Default Gateway:         192.168.2.254
Domain Name:             mycompany.com
DNS Server Search Order: 192.168.2.6
Domain Suffix Search Order:
```

hmcshutdown

The **hmcshutdown** command will shut down the Hardware Management Console. Before the system is shut down, console surveillance will also be disabled.

```
hmcshutdown -t { <number-of-minutes> | now } [ -r ] [-help ]
```

where:

- **-t** – the number of minutes to wait before halting or rebooting the system, or now.
- **-r** – reboot the console. If this option is not specified, the system will be halted.
- **-help** – prints a help message.

The following is an example of the usage of this command:

To reboot HMC in one minute:

```
[hmcusr@hmcproj hmcusr]$ hmcshutdown -t 1 -r
```

To halt HMC immediately:

```
[hmcusr@hmcproj hmcusr]$ hmcshutdown -t now
```

Ishmcusr

The **Ishmcusr** command is used to list the properties (names, role, and description) of HMC users:

```
Ishmcusr -u { ALL | <user-name> } [ -F <format> ] [ -help]
```

where:

- **-u** – the user name to be displayed. Valid values are:
 - **user-name** – to display information for a specified HMC user.
 - **ALL** – to display information for all valid HMC users.
- **-F** – if specified, a delimiter-separated list of property names to be queried. Valid values are **name**, **access**, **role**, and **description**. Specifying **access** will return the user's access group as a keyword, such as sysadmin, that can be used as input to a subsequent **mkhmcusr** or **chhmcusr** command. Specifying **role** will return the user's access group as a printable string, such as System Administrator. When this option is specified, the output will be a delimiter-separated list of the values requested.
- **-help** – prints a help message.

The following is an example of the usage of this command:

```
[hmcusr@hmcproj hmcusr]$ lshmcusr -u ALL
User Name  Roles                               Full Name
hscroot    System Administrator                HSC Super User
hmcusr     System Administrator                hmcusr
comproom   Service Representative              Computer Room
```

mkhmcusr

The **mkhmcusr** command will create a user on the Hardware Management Console:

```
mkhmcusr -u <user-name> -a <access-name> [-d "description" ] [ -help ]
```

where:

- **-u** – the user name to be created. The name must be between 1 and 32 characters long and not begin with a digit (0-9).
- **-a** – the access group name. Valid values are **op**, **advop**, **sysadmin**, **usradmin**, **svcrep**, and **viewer**.
- **-d** – a string containing the description associated with the user
- **-help** – prints a help message.

The following is an example of the usage of this command:

```
[hmcusr@hmcproj hmcusr]$ mkhmcusr -u hscpe -a svcrep -d "IBM Service
Representatives usereid"
```

```
Enter the new password for user hscpe: ****
```

```
Retype the new password for user hscpe: ****
```

```
[hmcusr@hmcproj hmcusr]$ lshmcusr -u hscpe
```

```
User Name  Roles                               Full Name
hscpe      Service Representative              IBM Service Representatives usereid
```

chhmcusr

The **chhmcusr** command will modify a user's property on the Hardware Management Console:

```
chhmcusr -u <user-name> -t [ name | passwd | access | desc] -v <new-
value> [ -help ]
```

where:

- **-u** – the user name to be modified.
- **-t** – the type of property to change. Valid values are **passwd** for changing a user's password, **access** for changing a user's access group name, **desc** for changing a user's description, and **name** for changing a user's name.
- **-v** – the new value of the property. When changing access group name, valid values are **op**, **advop**, **sysadmin**, **usradmin**, **svcrep**, and **viewer**. When changing the user's password, if no value is specified the user will be prompted to enter the password on the command line.
- **-help** – prints a help message.

The following is an example of the usage of this command.

To change the password of user hscpe to ibm:

```
[hmcusr@hmcproj hmcusr]$ chhmcusr -u hscpe -t passwd -v ibm
Enter the current password for user hscpe: ****
```

rmhmcusr

The **rmhmcusr** command will remove a user from the Hardware Management Console:

```
rmhmcusr -u <user-name> [ -help ]
```

where:

- **-u** – the user name to be removed.
- **-help** – prints a help message.

The following command removes user hscpe:

```
[hmcusr@hmcproj hmcusr]$ rmhmcusr -u hscpe
```

lscuod

The **lscuod** command will list information related to the Capacity Upgrade on Demand system features:

```
lscuod -m "managed-system" -r [ cpu | mem ] -t [ reg | order ] [ -F
<format> ] [ -help ]
```

where:

- **-m** – the name of the managed system for which CUoD information should be listed. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and in the form *mmmm-ttt*sssssss*, where *mmmm* is the machine type, *ttt* is the model type, and *sssssss* is the serial number of the managed system.
- **-r** – the type of resources to query the information. Valid values are **cpu** for processor and **mem** for memory.
- **-t** – the type of listing to display. Valid values are **reg** for regular CUoD resource information, or **order** for CUoD resource order information.
- **-F** – a delimiter-separated list of names representing the desired properties to query. Valid values are `capacity_card_ccin`, `capacity_card_id`, `capacity_card_serial_num`, `cpu_activated`, `cpu_capacity_cond`, `cpu_days_hours`, `cpu_entry_check`, `cpu_seq_num`, `immed_cpu`, `immed_mem`, `installed_cpu`, `installed_mem`, `mem_activated`, `mem_capacity_cond`, `mem_days_hours`, `mem_entry_check`, `mem_seq_num`, `not_perm_cpu`, `not_perm_mem`, `perm_cpu`, `perm_mem`, `resource_id`, `system_serial_num`, and `system_type`.
- **-help** – prints a help message.

My demo system does not have CUoD capabilities, and I obtain the following output when trying to check this:

```
[hmcusr@hmcproj hmcusr]$ lscuod -m '7038-6M2*C25699A' -r mem -t reg
The command entered is either missing a required parameter or a
parameter value is invalid. The required parameters for this command are
-c, -r, -m, and -t. Please check your entry and retry the command.
```

chcuod

The **chcuod** command will change the Capacity Upgrade on Demand attribute:

```
chcuod -o [ e | d | s ] -m "managed system" -k "activation key"
-q quantity -r [ cpu | mem ] [ -help ]
```

where:

- **-o** – the operation to perform. **s** indicates a set activation key operation, **d** indicates disabling activate immediately, and **e** indicates enabling activate immediately.
- **-m** – the name of the managed system where the CUoD attribute should be changed. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and in the form *mmm*ssss*, where *mmm* is the machine type and *ssss* the serial number of the managed system.
- **-k** – the activation key to be sent to the managed system.
- **-q** – the quantity of processors or memory to enable activate immediate.
- **-r** – the resource type to operate on. Valid values are **cpu** for processors or **mem** for memory.

Issyscfg

The **Issyscfg** command will list the system configuration:

```
Issyscfg -r [ sys | frame | alpar | lpar | prof | sysprof ]
-n object-name [ | -all ] [ -m "managed-system" ]
[ -p "partition-name" ] [ -F format | -z ] [ -help ]
```

where:

- **-r** – the system resource type to query. Valid values are **alpar** for affinity logical partition, **lpar** for partition, **prof** for profile, **sysprof** for system profile, **sys** for managed system, and **frame** for frame object.
- **-n** – the name of the object to perform the query on.
- **-all** – if specified, retrieve information for all resource specified. This argument cannot be used with the **-n** flag.
- **-m** – the name of the managed system where the resource

is configured. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and of the form *mmmm-ttt*sssssss*, where *mmmm* is the machine type, *ttt* is the model type, and *sssssss* the serial number of the managed system.

- **-p** – if the resource type is **prof**, the user-defined name of the partition where the resource will be queried.
- **-F** – a delimiter-separated list of property names to be queried. Valid values are: `activated_profile`, `affinity_capability`, `boot_mode`, `cage_number`, `cec_capability`, `csp_surveillance_policy`, `csp_version`, `cuod_capability`, `default_profile`, `desired_cpu`, `desired_io`, `desired_mem`, `dlpar_capability`, `frame_number`, `id`, `is_real_frame`, `lmb_size`, `maximum_cpu`, `maximum_mem`, `minimum_cpu`, `minimum_mem`, `mode`, `model`, `name`, `op_panel_value`, `op_panel_window_count`, `partition_profile`, `power_off_policy`, `required_io`, `runtime_capability`, `serial_number`, `service_authority`, `sfp_surveillance`, `small_rmo`, `sni_config_mode`, `sni_device_id`, `sni_windows`, `state`, `total_cpu`, `total_mem`, and `type`.
- **-z** – alternative display format. If specified, the attributes will be displayed in the form `attr=value` per line. This flag cannot be specified with the **-F**.
- **-help** – prints a help message.

The following is an example of the usage of this command.

To list the details of all managed systems:

```
[hmcusr@hmcproj hmcusr]$ lssyscfg -r sys -all -z
name=proj
state=Ready
model=7038-6M2
serial_number=C26699A
affinity_capability=0
cec_capability=131
runtime_capability=16
cuod_capability=0
```

```
power_off_policy=false
cage_number=
csp_surveillance_policy=20
csp_version=V4.0
mode=255
lmb_size=256
op_panel_value=LPAR. . .
```

Different allowable values for various fields are:

- `affinity_capability` – `affinity_LPAR` creation capability field: 0 for non-affinity-capable systems, 2 for 4-way affinity-capable systems, 4 for 8-way affinity-capable systems.
- `cec_capability` – the sum of the following values signifying various capabilities of managed systems: 1 for SMP capable systems, 2 for LPAR capable systems, 4 for NUMA capable systems, 64 for a system with an external BPA communication ability, 128 for the ability to support multi interfaces.

In our example this field is $131=128+2+1$, for support of multi interfaces, LPAR and SMP respectively.

- `runtime_capability` – the sum of the following values, signifying various runtime capabilities of the managed systems: 8 for ALPRA capable systems, 16 for DLPAR capable systems, 32 for message passing capable systems, 64 for system with CUoD ability

In our example this field is 16, for support of DLPAR.

To list all profiles for the partition `proja`:

```
[hmcusr@hmcproj hmcusr]$ lssyscfg -r prof -m proj -p proja -all
```

Name	BootMode	DesiredCPU	DesiredMEM	MaxCPU	MaxMEM	MinCPU	MinMEM
Cegetel Dev 1	2	2	4096	8	8192	1	2048
Cegetel D sms 3	2	2	4096	8	8192	1	2048
All FC Adpter 1	2	2	4096	8	8192	1	2048
Ceget Dev Srv 2	2	2	4096	8	8192	1	2048

mksyscfg

The **mksyscfg** command will create a hardware resource configuration:

```
mksyscfg -r [ alpar | lpar | prof | sysprof ] -m "managed- system"  
[ -p "partition-name" ] [ -f configuration-file | -i attribute-  
value ="value" ... ]  
[ -help ]
```

where:

- **-r** – the system resource type to create. Valid values are **alpar** for affinity logical partition, **lpar** for partition, **prof** for profile, and **sysprof** for system profile.
- **-m** – the name of the managed system where the resource is configured. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and of the form *mmmm-ttt*sssssss*, where *mmmm* is the machine type, *ttt* is the model type, and *sssssss* the serial number of the managed system.
- **-p** – if the resource type is **prof**, the user-defined name of the partition where the hardware resource will be removed.
- **-f** – the file containing the configuration information needed to create the resource object. The format of data in this file must be in the form:

```
attribute-name=attribute value
```

There must be no space either side of the '=' sign. Each attribute/value pair must be on separate lines.

- **-i** – this option allows a user to enter configuration information on the command line, instead of using a file. Data entered at the command line must follow the same format as the data in the configuration file.

For affinity logical partition resource type:

-name (String)

-profile_name (String)

-partition_type

Valid values are:

2 - 4 processor configuration

3 - 8 processor configuration

-desired_io (comma separated)

physical location code

-required_io (comma separated)

physical location code

-service_authority(0 - off, 1 - on)

-sfp_surveillance(0 - off, 1 - on)

-sni_config_mode (0 - basic, 1- advanced)

-boot_mode

Valid values are:

norm - normal

dd - diagnostic default boot list

sms - SMS

of - OpenFirmware OK prompt

ds - Diagnostic Stored Boot List

-small_rmo (0 - off, 1 - on)

For LPAR profile resource types:

-name (String)

-minimum_cpu (number)

-maximum_cpu (number)

-desired_cpu (number)

- minimum_mem (number of MBs)
- maximum_mem (number of MBs)
- desired_mem (number of MBs)
- desired_io (comma separated)
 physical location code
- required_io (comma separated)
 physical location code
- service_authority (0 - off, 1 - on)
- sfp_surveillance (0 - off, 1 - on)
- sni_config_mode (0 - basic, 1- advanced)
- sni_device_id (numbers comma separated)
- sni_windows (numbers that are a multiple of 16 and
comma separated)
- boot_mode

Valid values are:

- norm - normal
- dd - diagnostic default boot list
- sms - SMS
- of - OpenFirmware OK prompt
- ds - Diagnostic Stored Boot List

- small_rmo (0 - off, 1 - on)

For partition resource type:

- name (String)
- profile_name (String)
- minimum_cpu (number)

- maximum_cpu (number)
- desired_cpu (number)
- minimum_mem (number of MBs)
- maximum_mem (number of MBs)
- desired_mem (number of MBs)
- desired_io (comma separated) physical location code
- required_io (comma separated) physical location code
- service_authority(0 - off, 1 - on)
- sfp_surveillance(0 - off, 1 - on)
- sni_config_mode (0 - basic, 1- advanced)
- sni_device_id (numbers comma separated)
- sni_windows (numbers that are a multiple of 16 and comma separated)
- boot_mode

Valid values are:

norm - normal

dd - diagnostic default boot list

sms - SMS

of - OpenFirmware OK prompt

ds - Diagnostic Stored Boot List

- small_rmo (0 - off, 1 - on)

For system profile resource type:

- name

- partitions (comma separated, partition names must be unique)

-profile_names (comma separated, profile names must be unique)

-help – prints a help message.

chsyscfg

The **chsyscfg** command will change the hardware resource configuration

```
chsyscfg -r [ alpar | lpar | prof | sysprof ] -m "managed- system"  
        [ -p "partition-name" ] [ -f configuration-file |  
        -i attribute-value ="value" ... ] -n name [ -help ]
```

where:

- **-r** – the system resource type to modify. Valid values are **alpar** for affinity logical partition, **lpar** for partition, **prof** for profile, and **sysprof** for system profile.
- **-m** – the name of the managed system where the hardware resource is configured. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and of the form *mmm*ssss*, where *mmm* is the machine type and *ssss* the serial number of the managed system.
- **-p** – if the resource type is **prof**, the user-defined name of the partition where the hardware resource will be modified.
- **-f** – the file containing the configuration information needed to modify the resource object. Format of the data in this file must be of the form:

```
attribute-name=attribute value
```

There must be no space before or after the '=' sign. Each attribute/value pair must be on separate lines.

- **-i** – this option allows a user to enter configuration information on the command line, instead of using a file. Data entered at the command line must follow the same format of data in the configuration file.

For LPAR profile resource type:

- name (String)
- minimum_cpu (number)
- maximum_cpu (number)
- desired_cpu (number)
- minimum_mem (number of MBs)
- maximum_mem (number of MBs)
- desired_mem (number of MBs)
- desired_io (comma separated)
physical location code
- required_io (comma separated)
physical location code
- service_authority (0 - off, 1 - on)
- sfp_surveillance (0 - off, 1 - on)
- sni_config_mode (0 - basic, 1- advanced)
- sni_device_id (numbers comma separated)
- sni_windows (numbers that are a multiple of 16 and
comma separated)
- boot_mode

Valid values are:

- norm - normal
- dd - diagnostic default boot list
- sms - SMS
- of - OpenFirmware OK prompt
- ds - Diagnostic Stored Boot List

-small_rmo (0 - off, 1 - on)

For partition:

resource type:

-name (String)

-default_profile_name (String) For system

resource type:

-name (String)

-power_off_policy (Ø - off, 1 - on)

-csp_surveillance_policy (Ø - off, 1 - on) For

system profile resource type:

-name

-partitions

-profile_names

- **-n** – the name of the object whose attribute will be modified. If you change the attribute of a managed system, specify the managed system's name with this argument and omit the **-m** flag.
- **-help** – prints a help message.

The following command changes the user-defined name of a system profile from sysprof to sysprof1:

```
chsyscfg -r sysprof -m proj -n sysprof -i "name=sysprof1"
```

The following command changes various attributes of a partition profile. The '+' indicates that the value(s) should be added to the current attribute's value.

```
chsyscfg -r prof -m proj -p lpar1 -n prof -i minimum_cpu=3  
maximum_cpu=6 desired_io+="UP13*2-3" required_io+="UP13*2-4,UP13*2-7"
```

rmsyscfg

The **rmsyscfg** command will remove the hardware resource

configuration:

```
rmsscfcfg -r [ lpar | prof | sysprof | sys ] -n name [ -m "managed-  
system" ]  
          [ -p "partition-name" ] [ -help ]
```

where:

- **-r** – the system resource type to be removed. Valid values are **lpar** for partition, **prof** for profile, **sys** for managed system, and **sysprof** for system profile.
- **-n** – the user-defined name of the system object to be removed.
- **-e** – if the system resource type is **sys**, the frame in which the managed system is contained must be specified here.
- **-m** – the name of the managed system where the hardware resource is configured. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and of the form *mmm*ssss*, where *mmm* is the machine type and *ssss* the serial number of the managed system.
- **-p** – if the resource type is **prof**, the user-defined name of the partition where the hardware resource will be removed.
- **-help** – prints a help message.

The following command removes a partition:

```
rmsscfcfg -r lpar -m cec1 -n lp1
```

bkprofdata

The **bkprofdata** command will back up profile data from a managed system to a file:

```
bkprofdata -m "managed-system" -f <file-name> [ -help ]
```

where:

- **-m** – the name of the managed system where the profile data is located. If there are multiple managed systems with the same user-defined name, specify the managed system

name enclosed in double quotes and of the form *mmm***ssss*, where *mmm* is the machine type and *ssss* the serial number of the managed system.

- **-f** – the file that will contain the profile data. If the file name is not fully qualified, the file will be searched in `/var/hsc/profiles/<MTMS>/` where `<MTMS>` is the machine type and serial number of the managed system.
- **-help** – prints a help message.

The following command stores the system back-up in a local directory:

```
[hmcusr@hmcproj hmcusr]$ bkprofdata -m proj -f /home/hmcusr/backup
[hmcusr@hmcproj hmcusr]$ ls -l backup
-rw-r--r-- 1 root root 20464 Dec 12 23:15 backup
```

The following command stores a system back-up in the default directory:

```
[hmcusr@hmcproj hmcusr]$ bkprofdata -m proj -f backup01 [hmcusr@hmcproj
hmcusr]$ ls -l /var/hsc/profiles/7038-6M2*C25699A
total 60
-rw-r--r-- 1 root root 20464 Dec 12 23:16 backup01
-rw-r--r-- 1 root root 20464 Dec 6 11:00 backupFile
-rw-r--r-- 1 root root 20464 Dec 12 23:06 x
```

rstprofdata

The **rstprofdata** command will restore profile data for a managed system from a file:

```
rstprofdata -m "managed-system" -f <file-name> -l <restore-
type> [ -help ]
```

where:

- **-m** – the name of the managed system where the profile data is to be restored. If there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and of the form *mmm***ssss*, where *mmm* is the machine type and *ssss* the serial number of the managed system.
- **-f** – the file that contains the profile data. If the file name is

not fully qualified, the file will be searched in `/var/hsc/profiles/<MTMS>/` where `<MTMS>` is the machine type and serial number of the managed system.

- **-l** – the type of restore to be performed. Valid values are:
 - 1 – full restore from file.
 - 2 – merge current profile with back-up with the back-up having priority.
 - 3 – merge current profile with back-up with the managed system having priority.
 - 4 – initialize machine.
- **-help** – prints a help message.

This article will be concluded next month.

*Alex Polak
System Engineer
APS Israel*

© Xephon 2004

```
chsysstate -n proj -r sys -o on -c proj a -b std
```

The following command activates a system profile:

```
chsysstate -r sysprof -m proj -o on -n sysprof1
```

3.24 Issvcevents

Display the hardware Serviceable Events or HMC console events.

```
Issvcevents -t { hardware | console } [ -d <number-of- days-  
to-go-back> ]  
[ -m "managed-system" -s {ALL | sp | lpar}  
[ -p "partition-name" ]] [ -w {ALL | "switch-mtms" } ] [ -F  
<format> ] [ -help ]
```

- t The type of events to display. Valid values are hardware for Serviceable Events, or console for Console type events.
- d The number of days to go back and search for Serviceable Events. By default, 7 will be used.
- m The name of the managed system to display events from. If

there are multiple managed systems with the same user-defined name, specify the managed system name enclosed in double quotes and of the form mmm*ssss, where mmm is the machine type and ssss is the serial number of the managed system. This parameter is valid only for hardware (serviceable) events and cannot be specified with the -w parameter.

-s The source of the events to display. Valid values are:

sp - to display events from the Service Processor

lpar - to display events from the partitions

ALL - to display all events

This parameter is valid only with the -m parameter.

-p The name of the partition where the serviceable events come from. This parameter is valid only with the -s parameter.

-w The switch to display events from. Specify ALL to view events from all switches, or specify a single switch enclosed in double quotes and of the form mmmm*ssss where mmmm is the machine type, and ssss is the serial number of the switch. This parameter is valid only for hardware (serviceable) events and cannot be specified with the -m parameter.

-F If specified, a delimiter-separated list of property names to be queried. Valid values are name, time, callhome, calledhome, errorclass, description. When this option is specified, the output will be a delimiter-separated list of the values requested.

—help Prints help message.

The following command will display events from the console:

```
[hmcusr@hmcproj hmcusr]$ lssvcevents -t console
Earliest Timestamp      Description
12/11/03 06:36:29 AM  HSCE2014  UserName hscroot Virtual terminal has
been open on partition proj of lpar id 7038-6M2*C25333A of managed
system proj ;
12/11/03 06:35:53 AM  HSCE2174  User hscroot login was successful from
remote host hmc65001.mycompany.com with IP address 199.221.37.55.
```

12/06/03 11:00:27 AM HSCE2016 UserName hscroot Logical Partition proj b with ID 002 of managed system proj has been activated with profile prof;
12/06/03 11:00:26 AM HSCE2163 User hscroot: Completed activating partition.
12/06/03 11:00:05 AM HSCE2026 11264 of memory from managed system proj with ID of 7038-6M2*C25699A has been assigned to LPAR proj b with ID 2;

References

1. AIX 5L Version 5.2 AIX Installation in Partitioned Environment, SC23-4382. IBM Corporation,
2. Effective System Management Using the IBM Hardware Management Console for pSeries, SC24-7038, IBM Corporation
3. Planning for Partitioned-System Operations, SA38-0626, IBM Corporation
4. Site and Hardware Planning Information, SA38-0508, IBM Corporation
5. IBM Hardware Management Console for pSeries Installation and Operations Guide, SA38-0590, IBM Corporation
6. Electronic Service Agent for eServer pSeries User's Guide, LCD4-1060, IBM Corporation
7. PCI Adapter Placement reference, SA38-0538, IBM Corporation

Alex Polak
System Engineer
APS Israel

© Xephon 2004

AIX news

ASG has announced that a free demo version of its Trackbird IT asset management software is now available to download from www.ensignuk.com.

ASG-TrackBird provides a continuously updated hardware and software inventory by enabling the creation and deployment of multiple collection points.

Using ASG-TrackBird, networks are scanned to discover workstations, agents are deployed to inventory the workstations, and inventory progress is monitored. ASG-TrackBird indicates whether an agent is installed and active on a workstation and whether collection is complete. If no agent exists on a workstation, one can be deployed with just a few clicks.

The product supports AIX, Solaris, all Windows platforms for servers or workstations, OS/2, and MACs.

For further information contact:
ASG, 1333 Third Avenue South, Naples, FL 34102, USA.
Tel: (239) 435 2200.
URL: www.asg.com and www.ensignuk.com.

* * *

Veritas has announced Version 4.7 of Veritas Bare Metal Restore, its server restore software that runs on AIX 5.2, Red Hat Enterprise Linux 2.1, and Microsoft's Windows Server 2003.

The Bare Metal Restore software automates some of the tasks required to bring a failed server back to life. Primarily, the software handles operating system re-installs and hardware configuration on its own.

With 4.7, customers can now write their own scripts to complement the recovery process.

Users can set up tasks for restoring databases and other applications specific to their data centres. This software will then be restored along with the operating system and hardware configuration.

In addition, Veritas now supports 'dissimilar system restore' for Windows servers. This makes it possible to restore a Windows image on new hardware.

For further information contact:
Veritas, 350 Ellis Street, Mountain View, CA 94043, USA.
Tel: (650) 527 8000.
URL: <http://www.veritas.com/products/category/ProductDetail.jhtml?productId=bmr>.

* * *

Create!form has announced the release of Create!form Version 3.0 - PSE1 Edition. Certified by PeopleSoft for EnterpriseOne users.

Version 3.0 provides data formatting capabilities and functionality including: intelligent tables for formatting and management of repeating line item data; data calculation function for items such as page subtotals in repaginated documents; variable subforms that enable the dynamic utilization of variable images and data within the document; automated form templates; and queue-based server architecture.

Create!form runs on AIX/RS6000, Sun Solaris, and HP/UX.

For further information contact:
Create!form, 101A First Avenue, Waltham, MA 02451, USA.
Tel: (781) 487 6799.
URL: <http://www.createform.com/products/createformJDEadvantages.shtml>.



xephon