

## Virtualization Standards for Business Continuity: Part 3

This is the third of a series of articles defining the policies, guidelines, standards, and procedures that provide the foundation of a virtualized environment, thus enabling business continuity, disaster recovery, and high availability, with an emphasis toward Return On Investment (ROI).

The focus of this article is to define virtual adapter slot numbering standards used by the VIO servers and to discuss the placement of physical adapters within a pSeries frame or I/O drawer. These standards enable the administrator more easily manage and maintain the various I/O components provided by the VIO servers to the client LPAR's.

The standards and procedures detailed here are result of two years of implementation work refining best practices with respect to system administration, high availability, disaster recovery, and business continuity.

The previous article in this series provided a shell script that automates the process of building VIO servers according to the standards that are presented in this article. Utilizing a shell script to build the VIO servers ensures they are built in a consistent and reproducible manner. One of the most difficult tasks associated with managing a virtualized environment is creating and maintaining the association physical resources with their virtualized counterparts. The physical resources that must be managed include storage, networking, and serial devices. This article will concentrate on storage and networking.

A recommended configuration for a virtualized environment includes redundant VIO servers, which provide redundant access to storage and networking devices. This means that each VIO server will be configured with multiple physical adapters for access to storage and multiple physical devices with access to networking. Typically, each VIO server is configured with 4 (four) network devices, configured in an etherchannel, and two (2) or three (3) fiber channel devices providing access to storage.

At the time of this writing, the maximum network throughput of a VIO server is 3.5 gigabits/second, therefore implementing more than three (4) gigabit ethernet adapters cannot be fully utilized. The fourth adapter in the recommended configuration is usually implemented as a fail-over adapter for the etherchannel and is only utilized if the etherchannel becomes inoperable.

Configuration of the virtual SCSI adapters requires the knowledge of disk layouts as well as networking configuration. The virtual SCSI adapters require server and client side adapters to be configured on the HMC. The server side portion of the SCSI adapter, configured on the VIO, requires the definition of a frame wide unique "slot number". For high availability, the server side portion of the SCSI adapter must be configured on both VIO servers in a dual VIO configuration.

For each client LPAR that uses virtual disk or logical volumes, a client side virtual SCSI adapter must be configured on the HMC, one for each VIO server. The client side of the virtual SCSI adapter requires additional information and it's settings to correspond with the server side of the SCSI adapter. Coordinating the slot numbers defined here will make debugging and tracking of problems much easier and is highly desired.

To define a slot numbering standard for VIO client/server environments, the slot numbers should be divided on an even/odd basis, even numbered slots shall only be used on even numbered VIO servers, odd numbered slots shall only be used on odd numbered VIO servers. Thus providing an easy mechanism to determine which slot is served by which VIO server. This assumes the host naming standard described in a previous article in this series, was used to name the VIO servers.

### Virtualization Standards for Business Continuity: Part 3

To keep track of each slot and to provide a recognizable pattern to slot numbering, a range of slot numbers has been arbitrarily selected for assignment with virtual SCSI adapters. This standard specifies the range of slot numbers to be those between 10 and 499. Any slot numbered 500 and above is designated by this standard for assignment with virtual ethernet adapters.

Using the example VIO server host names from the previous article in this series (Table 3.1) , virtual SCSI slot numbers can be assigned based on the ultimate purpose of the storage attached to the virtual SCSI adapter, as shown in table 3.2:

<i>VIO Server Name</i>	<i>Description</i>	<i>Managed Frame Name</i>
<i>dalapvio00</i>	First VIO Server node on the frame	Server-9119-590-SN12A345B
<i>dalapvio01</i>	Second VIO Server node on the frame	Server-9119-590-SN12A345B

**Table 3.1: Example LPAR names for VIO Servers**

<i>Virtual SCSI Adapter Slot number</i>	<i>VIO Server Numbering (even/odd)</i>	<i>VIO Server (Example)</i>	<i>Purpose</i>
100	even	dalapvio00	Operating system storage
101	odd	dalapvio01	Operating system storage
102	even	dalapvio00	Database/Application storage
103	odd	dalapvio01	Database/Application storage
104	even	dalapvio00	Database/Application storage
105	odd	dalapvio01	Database/Application storage
106	even	dalapvio00	Database/Application storage
107	odd	dalapvio01	Database/Application storage
108	even	dalapvio00	Miscellaneous
109	odd	dalapvio01	Miscellaneous
110	even	dalapvio00	Operating system storage
111	odd	dalapvio01	Operating system storage
112	even	dalapvio00	Database/Application storage
113	odd	dalapvio01	Database/Application storage
114	even	dalapvio00	Database/Application storage
115	odd	dalapvio01	Database/Application storage
116	even	dalapvio00	Database/Application storage

### Virtualization Standards for Business Continuity: Part 3

<i>Virtual SCSI Adapter Slot number</i>	<i>VIO Server Numbering (even/odd)</i>	<i>VIO Server (Example)</i>	<i>Purpose</i>
117	odd	dalapvio01	Database/Application storage
118	even	dalapvio00	Miscellaneous
119	odd	dalapvio01	Miscellaneous
120	even	dalapvio00	Operating system storage
121	odd	dalapvio01	Operating system storage
122	even	dalapvio00	Database/Application storage
123	odd	dalapvio01	Database/Application storage
124	even	dalapvio00	Database/Application storage
125	odd	dalapvio01	Database/Application storage
126	even	dalapvio00	Database/Application storage
127	odd	dalapvio01	Database/Application storage
128	even	dalapvio00	Miscellaneous
129	odd	dalapvio01	Miscellaneous

**Table 3.2: VIO Server Slot Numbering**

Under this standard, virtual SCSI adapters that are assigned slot numbers ending in 0 or 1, are used for communication with operating system storage.

Virtual SCSI adapters that are assigned slot numbers ending in 2, 3, 4, 5, 6, or 7 are used for communication with application or database storage.

Finally, virtual SCSI adapters that are assigned slot numbers ending in 8, or 9 are used for communication with miscellaneous other storage.

The “mkviolpar” script from the previous article in this series, creates virtual SCSI adapters according to this standardized slot numbering scheme described above. It creates virtual SCSI adapters for all slots numbered between 10 and 499 for slots ending with numbers 0, 1, 2, 3, 8, and 9. The even numbered slots should be created on the even numbered VIO server, the odd numbered slots on the odd numbered VIO server. The purpose of segmenting the SCSI adapters on an even/odd basis is to provide the administrator with a mechanism to easily identify and maintain these resources. Using this standard, up to 49 client LPARs could be configured on any single frame. Slight modifications to this standard would allow as many client LPARs as desired to be configured. Table 3.3 shows an example slot numbering sequence for dual VIO servers providing virtual SCSI adapters for three client LPARs:

<i>Client LPAR Name</i>	<i>Client LPAR Slots</i>	<i>VIO Server dalapvio00</i>	<i>VIO Server dalapvio01</i>	<i>VIO Server Assignment to LPAR</i>

### Virtualization Standards for Business Continuity: Part 3

	<i>(from VIO)</i>	<i>Slot Number (even)</i>	<i>Slot Number (odd)</i>	
dalapora00	100/101	100	101	Operating System Storage
dalapora00	102/103	102	103	Database Storage
dalapora00	104/105	104	105	<i>Unused</i>
dalapora00	106/107	106	107	<i>Unused</i>
dalapora00	108/109	108	109	Paging
dalapora01	110/111	110	111	Operating System Storage
dalapora01	112/113	112	113	Database Storage
dalapora01	114/115	114	115	<i>Unused</i>
dalapora01	116/117	116	117	<i>Unused</i>
dalapora01	118/119	118	119	Paging
dalapora02	120/121	120	121	Operating System Storage
dalapora02	122/123	122	123	Application Storage
dalapora02	124/125	124	125	<i>Unused</i>
dalapora02	126/127	126	127	<i>Unused</i>
dalapora02	128/129	128	129	<i>Unused</i>

**Table 3.3: Client LPAR Virtual SCSI Adapter Slot Numbering**

This standardized method of assigning slot numbers to virtual SCSI adapter greatly enhances the administrators ability to build, modify, and maintain the client LPARs and VIO servers. The administrator can immediately recognize which VIO server is providing active (or inactive) paths to storage from the client LPAR and where problems exist. The administrator is also able to take specific virtual SCSI adapters offline for maintenance or reconfiguration without effecting storage attached to other virtual adapters. Again, the script “mkviolpar” creates the virtual SCSI adapters on each VIO server, based on the even/odd numbering scheme, however it is up to the administrator to assign these adapters to each client LPAR according to the standard described here.

Normally, LUNs from a storage array are assigned to the physical adapters associated with the dual VIO servers. An individual LUN would be assigned to both VIO servers through the virtual SCSI adapters and presented to a client LPAR. The Multi-Path I/O (MPIO) drivers on the client LPAR then recognize that a single LUN is presented through multiple virtual paths (one from each VIO Server) , and a single “hdisk” is discovered by the “cfgmgr” command.

A detailed procedure for assigning storage to the virtual SCSI adapters will be discussed in a later article.

The next section of this article will discuss a slot numbering standard for the virtual ethernet adapters

### Virtualization Standards for Business Continuity: Part 3

provided by dual VIO servers. As mentioned previously, any slot numbered 500 and above is designated by this standard for assignment with virtual ethernet adapters. The three digit slot number assigned to virtual ethernet adapters is segmented by each digit of the number, and each digit has a specific meaning as shown in table 3.3:

<i>First Digit</i>	<i>Second Digit</i>	<i>Third Digit</i>
5: Boot/Service 6: Standby 7: Backup 8: Management 9: Intra-frame	0: Gigabit 5: 10/100 9: Bus	0,2,4,6,8: Even numbered VIO Server 1,3,5,7,9: Odd numbered VIO Server

*Table 3.3: Ethernet Slot Number Components*

The first digit of the ethernet slot numbers associated with virtual ethernet adapters is used to represent a variety of adapter purposes. Those purposes include database and application service, boot, standby, backup, management, and intra-frame communications.

- 500-599: boot/service
- 600-699: standby
- 700-799: backup
- 800-899: management
- 900-999: intra-frame

The second digit of the three digit ethernet slot number sequence will represent the adapter speed. Gigabit cards will be represented by 0 thru 4, 10/100 adapters by 5 thru 8, and bus speed adapters by a 9. This digit may be reconfigured as necessary.

The third digit of the three digit ethernet slot number assigned to virtual ethernet adapters associated with the VIO Server named \*vio0 will end with an even or odd number, depending upon which VIO server it is configured, if it is configured on the even numbered VIO server, it will have an even number; on the odd numbered VIO server, an odd number.

For the example configuration, it is assumed the physical ethernet adapters configured in the VIO servers for use as communication interfaces to the “service” network, are all gigabit capable adapters. It is also assumed the physical ethernet adapters configured in the VIO servers for use as communication interfaces to the “management” network, are all 10/100 megabit adapters. Using this information, the slot numbers designated for use by these adapters are shown in table 3.4:

<i>VIO Server Slot Number</i>	<i>Adapter Type</i>	<i>Adapter Speed</i>	<i>VIO Server</i>
500	Boot/Service	Gigabit	dalapvio00
501	Boot/Service	Gigabit	dalapvio01
590	Boot/Service	BUS	dalapvio00

### Virtualization Standards for Business Continuity: Part 3

<i>VIO Server Slot Number</i>	<i>Adapter Type</i>	<i>Adapter Speed</i>	<i>VIO Server</i>
591	Boot/Service	BUS	dalapvio01
850	Management	10/100	dalapvio00
851	Management	10/100	dalapvio01
890	Management	BUS	dalapvio00
891	Management	BUS	dalapvio01

**Table 3.4: Three Digit Ethernet Slot Number**

An LPAR may have multiple adapters of a variety of types, for example an LPAR that provides database services and is a member of an HACMP cluster may have service, boot, standby, management, backup, and intra-frame virtual adapters as follows:

- Gigabit adapter used for boot/service address
- Gigabit adapter used for standby address
- 10/100 backup adapter
- 10/100 management adapter
- 2 intra-frame adapters

The “mkviolpar” script builds several standardized ethernet adapters. Each VIO server is configured with one (1) virtual adapter for use by the client LPARs to connect to the service network, and one (1) virtual adapter for use by the client LPARs to connect to the management network. The service network is assumed to handle the application and database network traffic. The management network is used by the system administrators for monitoring and management of the systems. The following table (table 3.5) shows four (4) adapters associated with each VIO server, the non-trunked adapters configured on slots 590, 591, 890, and 891 are used to assign IP addresses on their respective networks to the VIO server, this will be further explained in a later article.

<i>VIO Server Slot Number</i>	<i>VLAN ID</i>	<i>Adapter Type</i>	<i>Adapter Purpose</i>	<i>Adapter Speed</i>	<i>VIO Server</i>
500	500	Trunked	Boot/Service	Gigabit	dalapvio00
590	500	non-Trunked	Boot/Service	BUS	dalapvio00
850	850	Trunked	Management	10/100	dalapvio00
890	850	non-Trunked	Management	BUS	dalapvio00
501	501	Trunked	Boot/Service	Gigabit	dalapvio01
591	501	non-Trunked	Boot/Service	BUS	dalapvio01
851	851	Trunked	Management	10/100	dalapvio01
891	851	non-Trunked	Management	BUS	dalapvio01

## Virtualization Standards for Business Continuity: Part 3

*Table 3.5: Standardized Slot number / VLAN ID Association*

Finally, the virtual ethernet slot numbering standard can be propagated through to the client LPAR. Each client LPAR will be configured with a virtual ethernet adapter for each network from each VIO server. The example configuration discussed in this article would require two virtual ethernet adapters, one from each VIO server, for the “service” network (slots 500 and 501), and two virtual ethernet adapters, one from each VIO server, for the “management network (slots 850 and 851). The result of this example is that each client LPAR would be configured with four virtual ethernet adapters as shown in table 3.6:

<i>Client LPAR Name(example)</i>	<i>Client LPAR Slot Number</i>	<i>VIO Server Name(example)</i>	<i>VIO Server Slot Number</i>	<i>VLAN ID</i>
dalapora00	500	dalapvio00	500	500
dalapora00	501	dalapvio01	501	501
dalapora00	850	dalapvio00	850	850
dalapora00	851	dalapvio01	851	851
dalapora01	500	dalapvio00	500	500
dalapora01	501	dalapvio01	501	501
dalapora01	850	dalapvio00	850	850
dalapora01	851	dalapvio01	851	851
dalapora02	500	dalapvio00	500	500
dalapora02	501	dalapvio01	501	501
dalapora02	850	dalapvio00	850	850
dalapora02	851	dalapvio01	851	851

*Table 3.6: Client LPAR Ethernet Slot Numbering*

The purpose of this VIO server and client LPAR design is to provide automated fail over redundancy for the network connections as well as the storage connections. In a large data center, the administrator must be able to easily identify, track, monitor, and maintain the various resources utilized by all systems. It is also necessary for new personnel to become productive as soon as possible. Implementation of standards such as this ensure the ability to achieve these goals and contribute to the overall business continuity of the organization.

The previous article in this series promised this article would provide a web based reporting mechanism for identifying all physical and virtual adapters assigned to all VIO servers and client LPAR's. The length of this article prohibits the inclusion of the entire reporting mechanism at this

## Virtualization Standards for Business Continuity: Part 3

time, so only the data gathering portion will be included here, and the data processing portion will be presented in the next article of this series. The data gathering portion of this mechanism is a Korn Shell 93 script that communicates with the HMC via remote ssh commands. The HMC host name and user name is specified on the command line of this script. The data retrieved from the HMC is stored in a local directory, which can be specified by the user as a command line argument to the script. This script assumes that it has password-less ssh access to the HMC.

This HMC data gathering script is expected to be executed on a periodic basis through an automated scheduling system such as "cron". This data is critical to being able to *automatically* generate business continuity, disaster recovery, and high availability documentation, which should be a goal of all AIX system administrators. This script should probably be executed at least once a month, once a week would be better, once a day is probably too much. How often it is executed will depend upon your organization's business continuity requirements.

```
#!/usr/bin/ksh93
#####
function usagemsg_gethmcdata {
    print "
Program: gethmcdata

Retrieves frame, LPAR, adapter, and slot information
from an HMC.

Usage: ${1##*/} [-?vV] -h hmchostname [-u hmcuser] [-d /localdir]

Where:
-h = HMC host name (required parameter)
-u = HMC user name (default: hscroot)
-d = Local Directory in which to store HMC data (default: /tmp/hmcdata)
-v = Verbose mode - displays gethmcdata function info
-V = Very Verbose Mode - debug output displayed

Author: Dana French (dfrench@mtxia.com) Copyright 2007
\"AutoContent\" enabled
"
}
#####
####
#### Description:
####
#### This function retrieve a variety of information from an
#### HMC and stores that information in a file. This
#### information, stored as data, can then be processed to
#### create reports of any type.
####
#### Assumptions:
####
#### This function assumes that whatever HMC user name is
#### specified, that it has password-less access to the
#### specified HMC.
#### It is assumed the directory containing the "ssh" command
#### is in the PATH variable.
####
#### Dependencies:
####
#### This function is dependent upon the user specifying the
#### HMC host name on the command line. The HMC host name is
#### a required parameter.
####
#### This function is dependent upon the HMC being configured
#### to allow remote ssh login access.
```

## Virtualization Standards for Business Continuity: Part 3

```
#####
#### Products:
#####
#### This function produces data files that contain
#### information about the physical and virtual configuration
#### of the frame. These data files are, by default, stored
#### in the "/tmp/hmcddata" directory, or a directory
#### specified by the user.
#####
#### Configured Usage:
#####
#### The HMC host name is a required parameter, all others
#### are optional.
#####
####   gethmcddata -v -h hmchostname -u hscroot
#####
#### Details:
#####
#####
function gethmcddata {
  typeset VERSION="1.1"
  typeset TRUE="1"
  typeset FALSE="0"
  typeset EXITCODE="0"
  typeset VERBOSE="${FALSE}"
  typeset VERYVERB="${FALSE}"
  typeset HMCNAME=""
  typeset HMCUSER="hscroot"
  typeset SYSNAME=""
  typeset HMCLIST="${FALSE}"
  typeset HMCDATA="/tmp/hmcddata"
  typeset SYS
  typeset ARG
  typeset ARGS
  typeset ARYLINE
  typeset SYSCNT
  typeset HMC CNT
}

#####
#### Process the command line options and arguments, saving
#### the values as appropriate.
#####

while getopts ":vVu:h:d:" OPTION
do
  case "${OPTION}" in
    'h') HMCNAME="${OPTARG}";;
    'u') HMCUSER="${OPTARG}";;
    'd') HMCDATA="${OPTARG}"
        HMCDATA="${HMCDATA}%%+(/)";;
    'v') VERBOSE="${TRUE}";;
    'V') VERYVERB="${TRUE}";;
    '?') usagemsg_gethmcddata "${0}" && return 1 ;;
    ':') usagemsg_gethmcddata "${0}" && return 1 ;;
    '#') usagemsg_gethmcddata "${0}" && return 1 ;;
  esac
done

shift $(( ${OPTIND} - 1 ))

#####
#### Verify the HMC host name, if not display an error message and return
#### from this script.
#####

trap "usagemsg_gethmcddata ${0}" EXIT
if [[ ! "_${HMCNAME}" = "_" ]]
then
  print -u 2 "ERROR: HMC host name not specified"
  return 2
fi
```

## Virtualization Standards for Business Continuity: Part 3

```
fi
trap "-" EXIT

####
#### Display some program and parameter info
#### if "VERBOSE" mode was specified.
####

(( VERYVERB == TRUE )) && set -x
(( VERBOSE == TRUE )) && print -u 2 "# Version.....: ${VERSION}"
(( VERBOSE == TRUE )) && print -u 2 "# HMC Name.....: ${HMCNAME}"
(( VERBOSE == TRUE )) && print -u 2 "# HMC User.....: ${HMCUSER}"
(( VERBOSE == TRUE )) && print -u 2 "# Local Directory: ${HMCDATA}"

#####
####
#### Create the local directory in which to store the data files.
####
#####

mkdir -p "${HMCDATA}"
if ! cd "${HMCDATA}"
then
    print -u 2 "# ERROR: unable to cd to ${HMCDATA}"
    exit 1
fi

#####
####
#### Define the system level arguments to use with the HMC
#### command "lssyscfg".
####
#### "lssyscfg" parameter fields: r-value rsubtype level
####
#####

SYSCNT="0"
#         rvalue.rsubtype.level
SYSARGS[SYSCNT++]="lpar.null.null"
SYSARGS[SYSCNT++]="sys.null.null"
SYSARGS[SYSCNT++]="sysprof.null.null"

#####
####
#### Define the resource level arguments to use with the HMC
#### command "lssyscfg".
####
#### "lssyscfg" parameter fields: r-value rsubtype level
####
#####

HMCCNT="0"
#         rvalue.rsubtype.level
HWRESARGS[HMCCNT++]="mem.hmc.lpar"
HWRESARGS[HMCCNT++]="proc.hmc.lpar"
HWRESARGS[HMCCNT++]="io.bus.null"
HWRESARGS[HMCCNT++]="io.iopool.pool"
HWRESARGS[HMCCNT++]="io.iopool.sys"
HWRESARGS[HMCCNT++]="io.slot.null"
HWRESARGS[HMCCNT++]="io.taggedio.null"
HWRESARGS[HMCCNT++]="io.unit.null"
HWRESARGS[HMCCNT++]="mem.bus.lpar"
HWRESARGS[HMCCNT++]="mem.bus.sys"
HWRESARGS[HMCCNT++]="mem.eth.lpar"
HWRESARGS[HMCCNT++]="mem.eth.sys"
HWRESARGS[HMCCNT++]="mem.hsl.lpar"
HWRESARGS[HMCCNT++]="mem.hsl.sys"
HWRESARGS[HMCCNT++]="mem.iopool.lpar"
HWRESARGS[HMCCNT++]="mem.iopool.sys"
HWRESARGS[HMCCNT++]="mem.null.lpar"
```

## Virtualization Standards for Business Continuity: Part 3

```
HWRESARGS[HMCNT++]="mem.null.sys"
HWRESARGS[HMCNT++]="mem.scsi.lpar"
HWRESARGS[HMCNT++]="mem.scsi.sys"
HWRESARGS[HMCNT++]="mem.serial.lpar"
HWRESARGS[HMCNT++]="mem.serial.sys"
HWRESARGS[HMCNT++]="mem.slot.lpar"
HWRESARGS[HMCNT++]="mem.slot.sys"
HWRESARGS[HMCNT++]="mem.taggedio.lpar"
HWRESARGS[HMCNT++]="mem.taggedio.sys"
HWRESARGS[HMCNT++]="mem.unit.lpar"
HWRESARGS[HMCNT++]="mem.unit.sys"
HWRESARGS[HMCNT++]="mem.virtualopti.lpar"
HWRESARGS[HMCNT++]="mem.virtualopti.sys"
HWRESARGS[HMCNT++]="proc.bus.lpar"
HWRESARGS[HMCNT++]="proc.bus.pool"
HWRESARGS[HMCNT++]="proc.bus.sys"
HWRESARGS[HMCNT++]="proc.eth.lpar"
HWRESARGS[HMCNT++]="proc.eth.pool"
HWRESARGS[HMCNT++]="proc.eth.sys"
HWRESARGS[HMCNT++]="proc.hsl.lpar"
HWRESARGS[HMCNT++]="proc.hsl.pool"
HWRESARGS[HMCNT++]="proc.hsl.sys"
HWRESARGS[HMCNT++]="proc.iopool.lpar"
HWRESARGS[HMCNT++]="proc.iopool.pool"
HWRESARGS[HMCNT++]="proc.iopool.sys"
HWRESARGS[HMCNT++]="proc.null.lpar"
HWRESARGS[HMCNT++]="proc.null.pool"
HWRESARGS[HMCNT++]="proc.null.sys"
HWRESARGS[HMCNT++]="proc.scsi.lpar"
HWRESARGS[HMCNT++]="proc.scsi.pool"
HWRESARGS[HMCNT++]="proc.scsi.sys"
HWRESARGS[HMCNT++]="proc.serial.lpar"
HWRESARGS[HMCNT++]="proc.serial.pool"
HWRESARGS[HMCNT++]="proc.serial.sys"
HWRESARGS[HMCNT++]="proc.slot.lpar"
HWRESARGS[HMCNT++]="proc.slot.pool"
HWRESARGS[HMCNT++]="proc.slot.sys"
HWRESARGS[HMCNT++]="proc.taggedio.lpar"
HWRESARGS[HMCNT++]="proc.taggedio.pool"
HWRESARGS[HMCNT++]="proc.taggedio.sys"
HWRESARGS[HMCNT++]="proc.unit.lpar"
HWRESARGS[HMCNT++]="proc.unit.pool"
HWRESARGS[HMCNT++]="proc.unit.sys"
HWRESARGS[HMCNT++]="proc.virtualopti.lpar"
HWRESARGS[HMCNT++]="proc.virtualopti.pool"
HWRESARGS[HMCNT++]="proc.virtualopti.sys"
HWRESARGS[HMCNT++]="virtualio.eth.lpar"
HWRESARGS[HMCNT++]="virtualio.eth.sys"
HWRESARGS[HMCNT++]="virtualio.hsl.lpar"
HWRESARGS[HMCNT++]="virtualio.hsl.null"
HWRESARGS[HMCNT++]="virtualio.scsi.lpar"
HWRESARGS[HMCNT++]="virtualio.scsi.null"
HWRESARGS[HMCNT++]="virtualio.serial.lpar"
HWRESARGS[HMCNT++]="virtualio.slot.lpar"
HWRESARGS[HMCNT++]="virtualio.slot.slot"

#####

#### Loop through each system managed by the HMC.

for SYSNAME in $( ssh -x ${HMCUSER}@${HMCNAME} lssyscfg -r sys -F name )
do

    (( VERBOSE == TRUE )) && print -u 2 "# HMC Name.....: ${HMCNAME}"
    (( VERBOSE == TRUE )) && print -u 2 "# System Name....: ${SYSNAME}"

#### Create a data file to contain the HMC host name

    print -- "hmcname=${HMCNAME}" > ${HMCDATA}/${SYSNAME}.hmc.null.out
```

## Virtualization Standards for Business Continuity: Part 3

```
#### Retrieve the system level data from the HMC.

for ARYLINE in "${SYSARGS[@]}"
do
IFS=""
ARGS=( ${ARYLINE} )
IFS=$'\t\n'
[[ "_${ARGS[0]}" == "_null" ]] && ARGS[0]=" "
[[ "_${ARGS[1]}" == "_null" ]] && ARGS[1]=" "
[[ "_${ARGS[2]}" == "_null" ]] && ARGS[2]=" "
ARG="${ARGS[0]:+r ${ARGS[0]}} ${ARGS[1]:+--rsubtype ${ARGS[1]}} ${ARGS[2]:+--level ${ARGS[2]}}"

(( VERBOSE == TRUE )) && print -u 2 "# lssyscfg Args.: ${ARG}"
ssh -x ${HMCUSER}@${HMCNAME} "lssyscfg ${ARG} -m ${SYSNAME}" > ${HMCDATA}/${SYSNAME}.${ARYLINE}.out
done

#### Retrieve the resource level data from the HMC.

for ARYLINE in "${HWRESARGS[@]}"
do
IFS=""
ARGS=( ${ARYLINE} )
IFS=$'\t\n'
[[ "_${ARGS[0]}" == "_null" ]] && ARGS[0]=" "
[[ "_${ARGS[1]}" == "_null" ]] && ARGS[1]=" "
[[ "_${ARGS[2]}" == "_null" ]] && ARGS[2]=" "
ARG="${ARGS[0]:+r ${ARGS[0]}} ${ARGS[1]:+--rsubtype ${ARGS[1]}} ${ARGS[2]:+--level ${ARGS[2]}}"

(( VERBOSE == TRUE )) && print -u 2 "# lssyscfg Args.: ${ARG}"
ssh -x ${HMCUSER}@${HMCNAME} "lshwres ${ARG} -m ${SYSNAME}" > ${HMCDATA}/${SYSNAME}.${ARYLINE}.out
done

done
return 0
}

#####

gethmcdata "${@}"
```

Unfortunately, the IBM AIX version of Korn Shell 93 (ksh93), has a bug that causes this script not to run properly on an AIX system.. The bug prevents the “/” substitution operator from performing a marked replacement correctly. To run this script on an AIX system, download the official Korn Shell 93 AIX binary from [www.kornshell.com](http://www.kornshell.com) and it will resolve this issue. IBM has been notified of this bug, but as of the time of this writing, there is no resolution.

The Business Continuity policies, guidelines, standards, and procedures to be learned from this article are as follows:

### ***Policies:***

- Dual VIO Servers will be configured on each pSeries frame to provide redundancy and the ability to upgrade the VIO servers without client LPAR downtime.
- The maximum usable bandwidth of all physical ethernet adapters attached to any single VIO server shall not exceed 3.5 gigabits.
- Two or three HBA's shall be attached to each VIO server for redundant automated fail-over access to storage.
- Automated documentation generation shall be a goal of all system administration efforts.

## Virtualization Standards for Business Continuity: Part 3

### ***Guidelines:***

- VIO Server LPAR's should be created using a script to ensure consistency and adherence to standards.
- All HMC's should have remote access enabled for administration purposes.
- Two, three, or four gigabit ethernet adapters should be configured on each VIO servers to provide access for each network presented by the VIO.
- Two or three HBA's should be configured on each VIO server to provide access to storage.

### ***Standards:***

- This article provided standards for numerous parameters and settings associated with the creation of VIO Servers and client LPARs.

### ***Procedures:***

- The “mkviolpar” shell script provides a standardized, repeatable procedure for creating virtual SCSI and ethernet adapters on dual VIO servers across on multiple frames.
- The “gethmcdata” shell script provides an automated procedure for gathering data that identifies all physical and virtual adapters assigned to all VIO servers and client LPAR's.

The next article in this series will provide the data processing portion of the web based reporting mechanism for identifying all physical and virtual adapters assigned to all VIO servers and client LPAR's. The purpose of this mechanism is to provide an automated method of generating AIX system administration documentation as part of an organization's business continuity plan.

Dana French  
President, Mt Xia, Inc.  
<http://www.mtxia.com>  
615.556.0456