

Virtualization Standards for Business Continuity: Part 2

This is the second 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 a standardized and repeatable process for creating Virtual I/O Servers in the IBM pSeries environment.

The standards 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 detailed a naming standard for partitions, nodes, and host names. It also provided example names for VIO servers configured in a high availability arrangement. This arrangement consists of multiple pSeries frames, each configured with dual VIO servers, which in addition to high availability, also permits VIO server upgrades and maintenance without requiring client LPAR outages.

Dual VIO servers provide the client LPAR's with I/O redundancy in the event of failure of one of the VIO's. They also permit the system administrator to perform system maintenance on each VIO server without requiring an outage on any client LPAR.

Each client LPAR is subsequently configured to have access to the redundant resources provided by both VIO servers on the frame.

The example names used for the VIO servers are as shown in table 2.1:

<i>Partition Name</i>	<i>Description</i>	<i>Managed Frame Name</i>
<i>dalapvio00</i>	First VIO Server node on the first frame	Server-9119-590-SN12A345B
<i>dalapvio01</i>	Second VIO Server node on the first frame	Server-9119-590-SN12A345B
<i>dalapvio02</i>	First VIO Server node on the second frame	Server-9119-590-SN67D890E
<i>dalapvio03</i>	Second VIO Server node on the second frame	Server-9119-590-SN67D890E

Table 2.1: Example partition names for VIO Servers

To implement the VIO Server standards and ensure consistency across all frames providing virtualized LPARs, the creation of the VIO servers should be scripted. The following script allows the administrator to create two types of VIO Servers, a primary and a secondary. **The distinction between primary and secondary is arbitrary and only useful for discussing administration methodologies.** In operation there is no actual distinction. The administrative purpose of distinguishing between primary and secondary allows the administrator to identify the virtual devices by the VIO server to which they are attached. For this purpose, primary VIO servers are assumed to end with an even number (zero is regarded as an even number here), secondary VIO servers are assumed to end with an odd number. When the script creates the virtual devices associated with the VIO server, it uses even or odd number slots depending upon whether it is creating a primary VIO server or a secondary. When the script is complete, all the virtual devices on the primary VIO server will have been created using even numbered slots, and all the virtual devices on the secondary VIO server will have been created using

Virtualization Standards for Business Continuity: Part 2

odd numbered slots.

The “mkvioipar” script uses many values that can be modified to suit any environment, the values provided by the script have been found to be optimal for most implementations. These values are as described in table 2.2:

LPAR_ENV	When creating a VIO server LPAR, the value of this variable should be “vioserver”.
WORK_GROUP_ID	For the purpose of this article, this value will not be used and will have a static value of “-1”.
MIN_MEM	The minimum amount of memory the VIO Server LPAR will be allowed to have. When the LPAR boots, this amount of memory must be available or the LPAR will not boot. DLPAR operations will not be able to deallocate memory from the VIO Server LPAR below this value. The recommended initial value is “1024” MB.
DESIRED_MEM	When the VIO Server LPAR boots, it will have this amount of memory, if available on the frame. If not available, the LPAR will boot with as close to this amount of memory as it can allocate. The recommended initial value is “1024” MB.
MAX_MEM	The maximum amount of memory the VIO Server LPAR will be allowed to have. DLPAR operations will not be able to allocate more than this amount of memory to the VIO Server LPAR. The recommended initial value is “2048” MB.
PROC_MODE	The processor mode can be set to “dedicated” or “shared”. The recommended setting is “shared” which will allow unused processing units to be shared with other LPAR's.
MIN_PROC_UNITS	The recommended minimum processing units is “0.1” which is one tenth of a physical processor. At least “0.1” physical processing unit must be available in order for they VIO Server to boot, but since these LPAR's will normally be the first one's booted on the frame, this should never be a problem.
DESIRED_PROC_UNITS	The recommended setting for the desired processing units is also “0.1”.
MAX_PROC_UNITS	The recommended setting for the maximum processing units is “3.0”, which is 3 full physical processors. This setting is derived from the number of network adapter cards configured into the VIO Server. This will be discussed further in a later article.
MIN_PROCS	The minimum number of processors should be set to “1”.
DESIRED_PROCS	The desired number of processors should also be set to “1”.
MAX_PROCS	The maximum number of processors should be set to “3” to match the setting for the MAX_PROC_UNITS.
SHARING_MODE	The sharing mode should be set to “uncapped”.
UNCAP_WEIGHT	Use the default value of “128”.
BOOT_MODE	Use the value “normal”
CONN_MONITORING	Use the default value of “-1”.

Virtualization Standards for Business Continuity: Part 2

Table 2.2: VIO Server LPAR settings

These values can be changed by modifying the “mkviolpar” script or by defining environment variable and modifying those values. Example settings for environment variables are shown in table 2.3:

```
export LPAR_ENV="vioserver"
export WORK_GROUP_ID="-1"
export MIN_MEM="1024"
export DESIRED_MEM="1024"
export MAX_MEM="2048"
export PROC_MODE="shared"
export MIN_PROC_UNITS="0.1"
export DESIRED_PROC_UNITS="0.1"
export MAX_PROC_UNITS="3.0"
export MIN_PROCS="1"
export DESIRED_PROCS="1"
export MAX_PROCS="3"
export SHARING_MODE="normal"
export UNCAP_WEIGHT="128"
export BOOT_MODE="norm"
export CONN_MONITORING="-1"
```

Table 2.3: VIO Server Environment Variables

Note to Editor: Please insert the script called “mkviolpar” here.

The “mkviolpar” script uses the secure shell (ssh) as a communication mechanism with the HMC, which requires the HMC to be configured to allow “ssh” remote access.

Using the previously defined node naming standard, dual VIO servers can be defined on multiple pSeries frames using the “mkviolpar” script and the following command line examples:

```
# mkviolpar -v -h dallphmc01 -l dalapvio00 -u hscroot -s Server-9119-590-SN12A345B
# mkviolpar -v -h dallphmc01 -l dalapvio01 -u hscroot -s Server-9119-590-SN12A234B -S
# mkviolpar -v -h dallphmc01 -l dalapvio02 -u hscroot -s Server-9119-590-SN67D890E
# mkviolpar -v -h dallphmc01 -l dalapvio03 -u hscroot -s Server-9119-590-SN67D890E -S
```

Virtualization Standards for Business Continuity: Part 2

Where the name “dallphmc01” is used as the name of the Hardware Management Console, and “Server-9119-590-SN12A345B” is used as the system name of a p590 frame, and “Server-9119-590-SN67D890E” is the system name of a second p590 frame. The first command line example using the “mkviolpar” script creates the primary VIO server (the partition name ends with an even number). The second command line example creates a secondary VIO server (the partition name ends with an odd number and the “-S” option is specified). The default action of the “mkviolpar” script is to create a primary VIO server, meaning that all of the virtual devices created by the script end with an even number. The administration must use the “-S” option to cause the script to create virtual devices with an odd number. The third and fourth commands create the primary and secondary VIO servers on the second p590 frame.

The devices created for each VIO server by the “mkviolpar” script include virtual ethernet and SCSI adapters. The numbering scheme for these virtual devices is created according to a standardized methodology and will be discussed in a later article.

In order to support a business continuity mentality, the VIO servers require redundant physical devices. These devices include such adapters as ethernet, SCSI, and fiber channel. The details of these requirements will be discussed in a later article.

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.

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.

Standards:

- This article provided standards for numerous parameters and settings associated with the creation of VIO Servers.
- VIO Server host names shall comply with the naming standards.

Procedures:

- The “mkviolpar” shell script provides a standardized, repeatable procedure for creating VIO server LPAR's on multiple frames.

The next article in this series will discuss assigning physical I/O adapters to the VIO servers, and the policies, guidelines, standards, and procedures associated with these assignments. Also provided will be a web based reporting mechanism for identifying all physical and virtual adapters assigned to all VIO servers and client LPAR's.

Virtualization Standards for Business Continuity: Part 2

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