

FIBER CHANNEL TECH NOTE

TECHNICAL AND INFORMATIONAL
BN-0014-02



Fiber Channel Tech Note
BN-0014-02 (18-Feb-2019))

© 2019 AVT/Vere

vtAlpha and vtVAX are marketed jointly by AVT and Vere Technologies LLC

Table of Contents

1. Introduction	4
2. Fiber Channel	4
2.1. Fiber Channel Storage	4
2.2. Fiber Channel Adapter	5
2.2.1. Creating a virtual adapter	6
2.2.2. PCI slot	7
2.2.3. Configuring the KGPSA Adapter	8
2.3. Booting from Fiber Channel	10
2.4. LUN Map	10
2.5. Installing Tru64 on a Fiber Channel Disk	12
3. Tips and Tricks	13

1. Introduction

This document describes Fiber Channel specifics in a vtServer configuration. It is meant to bring a better understanding of Fiber Channel connections and behaviour in a vtServer (i.e. virtual) environment.

This document is related to the vtServer software, so only the Alpha (not the Integrity-platform) Fiber Channel specifics are referenced.

A basic understanding of the workings of Fiber Channel adapters and devices is assumed throughout this document. The configuration of the SAN and creation of devices on the SAN will not be discussed in this document.

2. Fiber Channel

2.1. Fiber Channel Storage

vtServer connects to multiple types of SAN storage (HP MSA, EVA, 3PAR, EMC, Hitachi, VPLEX, etc.) via modern host Fiber Channel (FC) adapters. Currently most QLogic and Emulex FC adapters are supported.

vtServer makes the attached SAN storage transparent for the Alpha Operating System (OpenVMS or Tru64), allowing it to connect SAN types for which no support is included in the OpenVMS or Tru64 Operating Systems.

In vtMonitor, the FiberChannel tab shows all physical FC adapters that are installed in the vtServer Host system. vtServer allows the sharing of physical FC adapters by multiple virtual Alpha systems.

Configuration	Storage	Toolbox	Logging	Online Manual
Host Storage	FibreChannel	iSCSI	NFS	SMB

When a FC adapter is shared, all storage attached to controllers connected to the FC is also shared. If this is not desirable, such as when storage zones are required, multiple virtual FC adapters may be created and attached to one physical FC adapter.

Each virtual FC adapter may be shared or not, as desired, and configured to access a subset of the physical storage.

n.b. To configure a FC adapter, right click on one of the adapters to display a context menu with available options.

In the example below, physical FC adapter fcad3 has two virtual FC adapters assigned (fcad5 and fcad8). By assigning a virtual FC adapter in your SAN using the IDs of the virtual storage adapter, it is possible to restrict access to portions of the storage network to certain virtual Alpha systems (Virtual Zoning).

FIBER CHANNEL TECH NOTE

Name: fcad5
Type: Virtual
Node Name: 5090-4560-3cdb-8393
Port Name: 5090-4561-3cdb-8393
Port State: Online
Port Type: NPIV VPORT
Speed: 2 Gbit
Info: Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux

Name	State	Port Name	Info
fcad1	Online	2100-00e0-8b8e-01f7	QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
fcad2	Online	2100-00e0-8b88-fff6	QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
fcad3	Online	1000-0000-c976-8e48	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad5	Online	5090-4561-3cdb-8393	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad8	Online	5b4e-1eb1-49be-bd43	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad4	Online	1000-0000-c976-8e49	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad6	Online	5b4e-1eb1-4ef5-84fe	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad7	Online	5b4e-1eb1-ec56-799c	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux

Below you see the (different) devices connected to the virtual adapters fcad5 and fcad8 in the SRM console. Both adapters were connected to 2 KGPSA's in this example.

```
P00>>> show dev
pga0.0.0.0.0          PGA0      KGPSA      5b4e-1eb1-49be-bd43  fcad8
gga996.11.0.0.0      GGA996   HP         HSV300                0953  sg23
dga502.11.0.0.0      DGA502   HP         HSV300                0953  sg25
dga503.11.0.0.0      DGA503   HP         HSV300                0953  sg26
pgb0.0.0.1.0         PGB0     KGPSA      5090-4561-3cdb-8393  fcad5
gga996.8.0.1.0       GGA996   HP         HSV300                0953  sg22
dga604.8.0.1.0       DGA604   HP         HSV300                0953  sg24
dga600.8.0.1.0       DGA600   HP         HSV300                0953  sg27
dga601.8.0.1.0       DGA601   HP         HSV300                0953  sg28
```

2.2. Fiber Channel Adapter

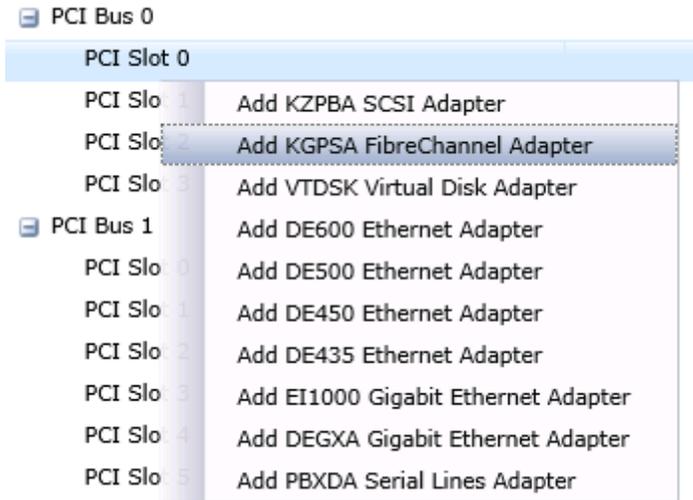
The Fiber channel adapter is called a *KGPSA* and it supports up to 32.767 devices.

In order to configure the adapter, a Fiber channel adapter must be present in the host X86 system you are using for the vtServer. This adapter will be shown in the FiberChannel tab under the Storage tab.

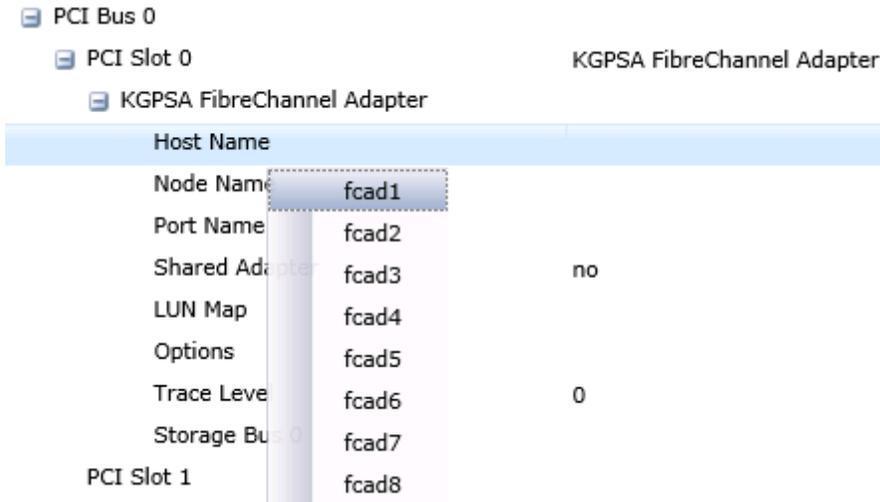
You have the choice to connect a *physical* FC adapter to your emulator or to connect a *virtual* FC adapter. n.b. How to create a virtual adapter will be explained in the next chapter.

To connect a physical or virtual FC adapter take the following steps:

- Select your emulator in the Configuration tab
- Select the PCI slot you want to connect the KGPSA adapter to



- Now, assign a FC adapter to the KGPSA
 - Right-click on the “Host Name”
 - A list of available adapters will be presented, choose the desired FC



2.2.1. Creating a virtual adapter

For you being able to add and configure a virtual Fiber channel adapter in an emulator configuration, you have to create it first (as a “child” of the physical adapter)

Follow these steps to create a virtual FC adapter:

- Open the vtMonitor browser interface
- Select the Storage tab
- Select the FiberChannel tab
- In the bottom half of the window, right-click on the physical FC-adapter

- Select “Add a Virtual FC adapter to ...”

Name	State	Port Name	Info
fcad1	Online	2100-00e0-8b8e-01f7	QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
fcad2	Add a Virtual FC adapter to fcad1		QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
fcad3	Online	1000-0000-c976-8e48	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad5	Online	5090-4561-3cdb-8393	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad8	Online	5b4e-1eb1-49be-bd43	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux

- A new sub-window will open, click “Add” to create the new virtual FC (it will receive a new and unique name)

Add a Virtual FC adapter to fcad1

Node Name:

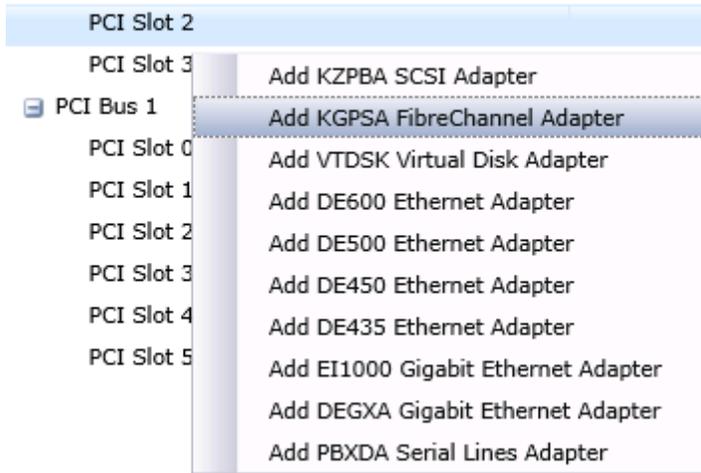
Port Name:

Name	State	Port Name	Info
fcad1	Online	2100-00e0-8b8e-01f7	QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
fcad2	Online	2100-00e0-8b88-fff6	QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
fcad3	Online	1000-0000-c976-8e48	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad5	Online	5090-4561-3cdb-8393	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux
fcad8	Online	5b4e-1eb1-49be-bd43	Emulex LPe11002-E FV2.82A4 DV11.0.0.0. HN:venus OS:Linux

n.b. certain types of FC adapters do not support the creation of a virtual adapter. If this is the case, an error-message like “*adapter fcadx does not support virtual ports*” will be shown after you click the “Add”-button

2.2.2. PCI slot

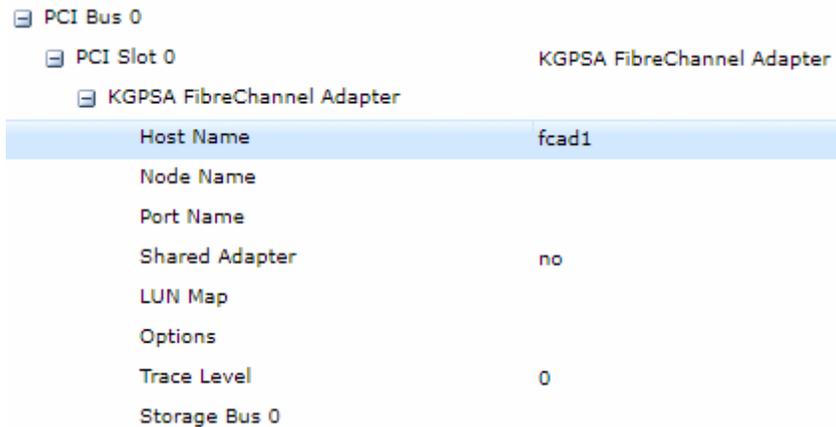
In vtMonitor, position the mouse pointer over a PCI slot, of a configuration, and right-click to display a context sensitive menu with options to add or remove the KGPSA Fiber Channel adapter to/from the slot.



n.b. If the emulator is running you have to stop and start the emulator to make it active.

2.2.3. Configuring the KGPSA Adapter

The virtual KGPSA adapter connects to the Fiber Channel storage elements or SAN that need to be connected to the virtual Alpha.



When a host name is assigned (fcad1 in KGPSA in PCI slot 0, example above), the SAN storage controllers will present the storage configuration to vtServer and changes to the configuration will be detected automatically.

- When the host name is defined, devices may not be manually configured. Previously configured devices will be retained in the configuration file but will be ignored. This is the preferred method for configuring SAN storage.
- *When the host name is omitted, each storage device to be accessed must be manually defined.*

The following types of storage devices are supported:

- Physical disks. Each accessible LUN on the SAN is available as a device on the vtServer host and as a virtual disk on the vtAlpha guest.

FIBER CHANNEL TECH NOTE

- Logical disks. Container files located on the SAN may be configured as logical disks which are presented to the vtAlpha guest OS as individual devices.
- Physical tapes. Direct attached tape device.

The virtual KGPSA adapter has 7 parameters that can be modified:

- *Host Name*: Name of the Fiber Channel adapter; when assigned the adapter will manage the available storage elements.
- *Node Name*: Node hardware address of the FC adapter.
- *Port Name*: Port hardware address of the FC adapter.
- *Shared Adapter*: Set to yes when this adapter is shared with another virtual Alpha.
- *Options*: Special options, to be used under guidance of your vtAlpha Support Organization
- *Trace Level*: To be used for debugging purposes as requested by your vtAlpha support organization; should be 0 for production situations.
- *LUN Map*: Mask i.e. hide devices and change device ID's
 - Format:
[portid=xxxxxx][productid=xxxxxx][,default][,lun:id],[lun:id,...][;portid=xxxxxx...]
In one of the following chapters the Lun Map will be explained more extensively.

The Node Name of the destination Fiber Channel adapter can be found in the Host tab under the Toolbox tab, and under the FiberChannel tab under Storage, in vtMonitor .

Example:

Fiber channel devices:

Controller: QLA200 FW:v3.03.28 DVR:v8.07.00.26-k
Host name: fcad1 (host5)
Node name: 2000-00e0-8b8e-01f7
Port name: 2100-00e0-8b8e-01f7
Port id: d20a00
Port type: NPort (fabric via point-to-point)
Speed: 2 Gbit
State: Online

Enter the Host name (fcad1 in this example) or the unique Node name (2000-00e0-8b8e-01f7) in the Host name field of the virtual Alpha configuration specifications.

SAN storage configuration changes will be automatically detected by the vtServer host. OpenVMS does not support automatic storage reconfiguration: the SYSMAN AUTOCONFIGURE command must be used to make the changes visible to the OpenVMS operating system. This is an OpenVMS limitation that applies to both physical and virtual Alpha configurations.

The devices/disks will be visible in the Alpha's SRM console with the following command.

P00>> show device

n.b. the customer is responsible for creating the LUN's

2.3. Booting from Fiber Channel

When using OpenVMS the use of the *wwidmgr* command is **not** needed.

Since a device list may become large with Fiber Channel configurations the console will now allow a matching device name to be given for the 'show device' command:

```
P00>>> show device
fga0.0.0.0.1 FGA0 KGPSA 1000-0000-c92b-116c
gga998.10.0.0.1 GGA998 HP MSA CONTROLLER 7.20 sg58
dga100.10.0.0.1 DGA100 HP MSA VOLUME 7.20 sg59
dga101.10.0.0.1 DGA101 HP MSA VOLUME 7.20 sg60
```

```
P00>>> show device f
fga0.0.0.0.1 FGA0 KGPSA 1000-0000-c92b-116c
fgb0.0.0.1.1 FGB0 KGPSA 1000-0000-c93a-389c
fgc0.0.0.2.1 FGC0 KGPSA 2100-00e0-8b1c-d44a
```

To setup a boot or dump device on the console, multiple paths may be given so that at boot time all devices are tried, and malfunctioning paths are skipped.

For example, to setup boot for dga100

```
P00>>> set bootdef_dev dga100.10.0.0.1,dga100.10.1.0.1,dga100.8.0.1.1,dga100.8.1.1.1
```

In this case we setup four paths which will be tried one at a time if we boot.

To boot from a specific path we can also give the complete path to the boot command:

```
P00>>> boot dga100.10.1.0.1
```

If we do not give a path but just the device name (*P00>>> boot dga100*), the console will attempt to boot from the first device it finds.

2.4. LUN Map

It is possible to change the mapping for lun's to a certain identifier for FC controllers that do not have the ability to setup a device identifier, which is normally required for VMS or Tru64. It is also possible to "un-configure" a certain lun so that it will not be used by an emulator.

For this we have the 'LUN Map' parameter in the KGPSA configuration.

n.b. a LUN Map change has preference over the "OS unit id".

The string can be specified like this:

```
[portid=xxxxxx][productid=xxxxxx][,default][,lun:id],[lun:id,...][;portid=xxxxxx...]
```

Selection can be limited to either a product id or a port id, or to one or more lun's.

An example makes this more clear. If the LUN Map is not specified then the emulator's log file will show something like this:

```
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 node 50060b00008247c0 port 50060b00008247c1 id 010000 luns 4
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 0 identifier 998 dev /dev/sg5 ident MSA CONTROLLER
```

FIBER CHANNEL TECH NOTE

```
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 1 identifier 1 dev /dev/sg6 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 2 identifier 2 dev /dev/sg7 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 3 identifier 3 dev /dev/sg8 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 node 50060b00008247c0 port 50060b00008247c9 id 010100 luns 4
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 0 identifier 999 dev /dev/sg17 ident MSA CONTROLLER
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 1 identifier 1 dev /dev/sg18 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 2 identifier 2 dev /dev/sg19 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 3 identifier 3 dev /dev/sg20 ident MSA VOLUME
```

In the above example we have one virtual KGPSA controller connected to a real Fiber channel controller (host4). There are two ports available (id 010000 and 010100) and each port has one controller, each giving access to 4 lun's. Lun 0 is always the controller itself and cannot be remapped.

If we want to un-configure lun 2 we can enter '2:0' in the lun map which will remove lun 2 from all controllers on this interface. If we just want to un-configure lun 2 on port 010100 then we can enter 'portid=010100,2:0' which will restrict the removal to that port. If we want to un-configure lun 2 where the ident is 'MSA VOLUME' we can enter 'productid=MSA VOLUME,2:0'.

The keyword 'default' means that there will be a one to one correspondence between the lun and the identifier. So lun 1 will get identifier 1, lun 2 gets identifier 2, etc.

Examples:

```
portid=10100,default,1:34,3:48;portid=10000,2:0
```

This means that controller with id 0x10100 will map all lun's one on one to an identifier (lun 1 = id 1, lun 2 = id 2, etc.). After that an optional list of mappings may follow. In this case lun 1 will be mapped to identifier 34, and lun 3 to id 48. The latter 2 mappings will overrule the default map. For port 0x10000 we will un-configure lun 2.

So we would get this:

```
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 node 50060b00008247c0 port 50060b00008247c1 id 010000 luns 3
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 0 identifier 998 dev /dev/sg5 ident MSA CONTROLLER
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 1 identifier 1 dev /dev/sg6 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 3 identifier 3 dev /dev/sg8 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 node 50060b00008247c0 port 50060b00008247c9 id 010100 luns 4
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 0 identifier 999 dev /dev/sg17 ident MSA CONTROLLER
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 1 identifier 34 dev /dev/sg18 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 2 identifier 2 dev /dev/sg19 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 3 identifier 48 dev /dev/sg20 ident MSA VOLUME
```

Notice that the above mapping may not make a lot of sense, but you get the idea.

Example:

```
productid=A6218A,default
```

Overrule the lun's as above but then for the device with the SCSI product identification A6218A.

Example:

```
1:0,3:123
```

This means, do not map lun 1 (that one will not be configured on this controller and will be invisible) and map lun 3 to identifier 123. Do this for all ports.

This is the result:

```
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 node 50060b00008247c0 port 50060b00008247c1 id 010000 luns 3
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 0 identifier 998 dev /dev/sg5 ident MSA CONTROLLER
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 2 identifier 2 dev /dev/sg7 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 0 lun 3 identifier 123 dev /dev/sg8 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 node 50060b00008247c0 port 50060b00008247c9 id 010100 luns 3
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 0 identifier 999 dev /dev/sg17 ident MSA CONTROLLER
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 2 identifier 2 dev /dev/sg19 ident MSA VOLUME
emulator.typhoon.PCHIP1.0.kgpsa : adp host4 ctrl 1 lun 3 identifier 123 dev /dev/sg20 ident MSA VOLUME
```

n.b. the “LUN Map” is per controller, so other controllers may get another mapping.

This way it is possible to fine-tune access to several lun’s from several different emulators on one host, something which is normally done with san-zoning (which will not work inside one host system with multiple emulators).

Here are some other examples of its use:

```
portid=10300,default,1:34,5:48
```

This means that for controller with id 0x10300 we will map all LUNs one on one to an identifier (lun 1 = id 1, lun 2 = id 2, etc.). After that the optional list of mappings follows.

In this case, lun 1 will be mapped to id 34, and LUN 5 to id 48.

The latter 2 mappings will overrule the default mapping.

Example:

```
productid=A6218A,default,1:34,5:48
```

This overrules the LUNs for the device with the SCSI product identification A6218A.

Example:

```
productid=HSV300,3:3,2:5
```

This definition will map dga3 to dga3 and dga2 to dga5

2.5. Installing Tru64 on a Fiber Channel Disk

Some special handling is required when installing Tru64 on a Fiber Channel disk. For this the console has the `wwidmgr` utility embedded, just like a real system. Before installing Tru64 on a Fiber Channel disk, the disk needs to be configured with the `wwidmgr` before it is recognized.

`wwidmgr` supports the following commands:

```
wwidmgr -clear
```

This will erase all previous settings and starts with a clean configuration.

```
wwidmgr -show adapter
```

This will show all Fiber Channel adapters on the virtual system:

Example:

```
P00>>> wwidmgr -show adapter
```

FIBER CHANNEL TECH NOTE

item	adapter	WWN	Topo
[0]	pga0.0.0.0.0	5090-4560-3cdb-8393	FABRIC

wwidmgr -show wwid

This will show all Fiber Channel devices that can be used for installation:

P00>>> wwidmgr -show wwid

[0] udid:604 wwid:01000010:6005-08b4-0009-9734-0001-b000-000c-0000 (ev:wwid0)

[1] udid:600 wwid:01000010:6005-08b4-0009-9746-0001-4000-01c8-0000 (ev:wwid1)

[2] udid:601 wwid:01000010:6005-08b4-0009-9746-0001-d000-000e-0000 (ev:none)

If installation on the device with udid 105 is desired then the following needs to be done:

P00>>> wwidmgr -quickset -udid 105

After this command it will be known to the console.

n.b. udid stands for “unit device identifier” - this is a nonnegative integer that is used in the creation of the OpenVMS device name. For example, udid:102 will be device DGA102

P00>>> wwidmgr -show wwid

[0] udid:100 wwid:01000010:6008-05f3-0006-aa30-a01d-7565-96a1-000f (ev:none)

[1] udid:101 wwid:01000010:6008-05f3-0006-aa30-a01d-7573-f10a-0010 (ev:none)

[2] udid:102 wwid:01000010:6008-05f3-0006-aa30-a01d-757b-6d40-0011 (ev:none)

[3] udid:103 wwid:01000010:6008-05f3-0006-aa30-a01d-7586-c4fb-0012 (ev:none)

[4] udid:104 wwid:01000010:6008-05f3-0006-aa30-a01d-758d-34eb-0013 (ev:none)

[5] udid:105 wwid:01000010:6008-05f3-0006-aa30-abe9-c1bd-40cf-0016 (ev:wwid0)

Notice the **ev** field, which indicates that the device is now known in the wwid0 console variable.

P00>>> show wwid0

wwid0 105 1 WWID:01000010:6008-05f3-0006-aa30-abe9-c1bd-40cf-0016

At the same time this will initialize the console variables N1 and N2 which are holding the port names of the controller for that device:

*P00>>> show n**

N1 500805f30006aa31

N2 500805f30006aa39

The wwidmgr settings will be saved in the *nvr* file.

n.b. On real systems the wwidmgr commands need to be followed by an *init* command. On vtAlpha this is not needed.

3. Tips and Tricks

- OpenVMS does not know or handle LUN's that are larger than 255
 - The solution is to add the desired number in the identifier field and add a number *other than 0* in the LUN field

FIBER CHANNEL TECH NOTE

KGPSA FibreChannel Adapter	
Host Name	host05
Node Name	
Port Name	
Shared Adapter	no
LUN Map	
Options	
Trace Level	0
Storage Bus 0	
New Logical Disk	Logical Disk
LUN	1
Identifier	356
Trace Level	0
Logical Disk	

