Abstract

This white paper discusses how to configure an EMC® Symmetrix® array in a Linux on System z environment.

October 2009
# Table of Contents

**Executive summary** ................................................................................................................. 4
**Introduction** ............................................................................................................................... 4  
  Audience .................................................................................................................................. 5  
  Terminology ............................................................................................................................... 5  
**Linux on System z connectivity** ......................................................................................... 5  
  Linux on System z disk options ............................................................................................... 6  
**FCP connectivity example** .............................................................................................. 6  
  Symmetrix FCP requirements ................................................................................................. 7  
  Adding FCP devices to Linux on System z .............................................................................. 8  
  Building the Linux on System z FBA device relationship .................................................... 8  
  z/VM IOCP example .................................................................................................................. 9  
  Linux on System z z/VM directory entry example .................................................................. 10  
  Querying FCP devices ............................................................................................................... 10  
  Bringing FCP devices online .................................................................................................... 11  
  Determining the WWPN – FCP .............................................................................................. 12  
  Configuring FCP devices on Linux on System z ..................................................................... 13  
    VM console message ............................................................................................................. 13  
    Partition a SCSI device via fdisk ......................................................................................... 14  
    Creating a filesystem ............................................................................................................ 15  
**ECKD connectivity example** .......................................................................................... 15  
  ECKD test environment .......................................................................................................... 15  
  Adding ECKD devices to Linux on System z .......................................................................... 16  
  Building the Linux on System z ECKD device relationship ................................................ 16  
  IOCDS example for Symmetrix – CKD .................................................................................... 18  
  Linux on System z VM directory entry example ................................................................... 18  
  Viewing ECKD DASD on z/VM ............................................................................................. 19  
  Configuring ECKD devices to Linux ...................................................................................... 20  
    Bringing the ECKD device online ....................................................................................... 20  
    Format ECKD devices .......................................................................................................... 20  
**DASDFMT under z/VM** ...................................................................................................... 22  
**Multipathing** .................................................................................................................. 23  
**EMC Solutions Enabler for Linux on System z** ................................................................. 24  
  EMC Solutions Enabler introduction .................................................................................... 24  
  Solutions Enabler ECKD requirements .................................................................................. 24  
**Conclusion** ....................................................................................................................... 26  
**References** ......................................................................................................................... 26
Executive summary

EMC® Symmetrix® storage arrays incorporate the industry’s leading technology components into the world’s most advanced storage architecture, delivering the highest levels of performance and throughput available. The Symmetrix series of arrays provide the greatest flexibility for integrating into most enterprise data center environments.

As data centers continue to look at ways to increase their workload capacity and performance while reducing their power, cooling, and floor space costs, virtualization technologies are fast becoming a critical element in most production environments. The IBM System z environment offers the ability to consolidate physical servers into virtual machines, reducing the physical footprint, power, and management challenges associated with many physical servers.

Linux on System z is an encompassing term used to describe Linux versions created to run on IBM’s System z architecture. The mainframe offers predictability — reliability, standards, and consistency; Linux offers flexibility and innovation. Since a Symmetrix storage array can support both environments with ease, it’s a natural evolution to bring them together to get the best enterprise-class solution available.

z/VM brings the strength of a mature virtualization to open systems enterprise computing. z/VM provides the ability to remove the physical limitations of individual servers through virtual machines while providing the ability to centrally manage and monitor the environment. Its consistency, along with the innovative, open-source, community-driven operating system of Linux, makes this a match made in cyber heaven. Linux on System z brings the best of both worlds together. The combination provides stability, high-performance data access, overall system performance, and scalability of the mainframe environment coupled with the flexibility of Linux on System z.

Consolidation is one of the primary reasons customers are moving to the Linux on System z environment. The green initiative has had an impact in many areas of the data center. Everything is at a premium these days – power, space, physical and human resources, and so on. High utilization is key to success. Everyone wants to better utilize their inherent knowledge base and personnel, and Linux on System z can assist with this endeavor. Moving to Linux on System z helps customers physically reduce the number of servers, footprint, power usage, HVAC, and the most valuable resource — personnel. The infrastructure that z/VM provides for scheduling, automation, performance monitoring, and reporting of virtual machines running Linux on System z are just a few of the productivity benefits that come with running Linux virtual machines under z/VM.

There are several Linux on System z host options that are Linux, distributed by either Red Hat or Novell’s SUSE, and are capable of running on the System z mainframe. Red Hat and SUSE can run standalone in their own Logical Partition (LPAR) or run in a virtual machine under z/VM as a guest operating system.

Most often, customers run Linux on System z native in an LPAR for large, intensive application environments. Some run Linux on System z when z/VM is not deployed in their environment. The most common environment seen so far is Linux on System z in a virtual machine (under z/VM). This offers the most flexibility with configuration, easy expansion, sharing of resources, and the ability to run many guest Linux on System z virtual machines. With any complex configuration there are always challenges. Part of the challenge for this environment is the union of mainframe and Linux – each with their own taxonomies (especially when it comes to storage). Terminology adds another challenge; the mainframe uses DASD for ECKD/CKD, open systems uses Logical Unit Number (LUN), Symmetrix uses Symmetrix Logical Volume (SLV), others use device or volume, and z/VM has an additional disk construct called a minidisk (mdisk). No matter what we call storage — DASD, device, disk, minidisk — all are abstractions of storage whether they be full, partial, or multiple physical devices.

Introduction

This white paper documents considerations needed to deploy Linux on System z with a Symmetrix storage array. It will examine both ECKD and FBA device configuration from the Symmetrix, to the z/VM platform storage allocation and configuration, concluding with device enablement to Linux on System z.
within this paper it will also be referred to as Linux). It describes the various drive emulations used to optimize a Linux on System z environment while discussing the basic configuration tasks needed to operate with a Symmetrix array. This consists of explaining the FBA and ECKD relationship from the Symmetrix to z/VM, to Linux on System z running in a guest virtual machine. As there is a lot of flexibility when defining disk devices to Linux on System z, the focus of this paper is on ECKD and SCSI FBA direct-attached devices.

In this paper the terminology of direct-attached, attached, or dedicated devices is discussed from the z/VM and guest virtual machine standpoint. This terminology has no bearing on the physical connectivity of the environment. Examples for each environment are examined along with the steps needed to define them to Linux on System z.

In this paper we will examine a Linux on System z environment running as a guest under z/VM in an LPAR on an IBM System z10.

**Audience**

This white paper is intended for technology professionals, system architects, and IT administrators or technical staff interested in how to configure Linux on System z for Symmetrix environments. This white paper assumes prior basic knowledge of storage concepts and management, Symmetrix operating environments, and especially z/VM and Linux concepts and terminology.

**Terminology**

The terminology in this paper crosses both mainframe and open system disciplines. This section is provided to explain different acronyms used in this white paper.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECKD</td>
<td>Extended Count Key Data, a mainframe storage format. Its physical block is the smallest unit.</td>
</tr>
<tr>
<td>FBA</td>
<td>Fixed Block Architecture, a disk layout where each physical record is the same size.</td>
</tr>
<tr>
<td>FICON</td>
<td>Fibre Connectivity; a high-speed I/O interface that connects storage devices to mainframe computers.</td>
</tr>
<tr>
<td>FCP</td>
<td>Fibre Channel Protocol; a storage area network connectivity protocol.</td>
</tr>
<tr>
<td>IFL</td>
<td>Integrated Facility for Linux, a type of processing unit available on the System z that allows additional central processors to be dedicated to Linux on System z.</td>
</tr>
<tr>
<td>SCSI disk</td>
<td>Small Computer System Interface disk, a set of standards defining connectivity to computer systems. Used commonly with open systems environments to reference FBA disks.</td>
</tr>
<tr>
<td>WWNN</td>
<td>World Wide Node Name</td>
</tr>
<tr>
<td>WWPN</td>
<td>World Wide Port Name</td>
</tr>
<tr>
<td>z/VM</td>
<td>An IBM operating system designed for virtual machines. In the mainframe environment, the OS allows one or more guests to run within each virtual machine under z/VM.</td>
</tr>
</tbody>
</table>

**Linux on System z connectivity**

To connect a Symmetrix array to a Linux on System z environment, you need to choose a channel and disk type. Fibre Connectivity (FICON) and Fibre Channel Protocol (FCP) are the available connectivity choices. The channel type determines the disk type of either ECKD or FBA. FICON is a follow-on protocol to ESCON supporting full duplex data transfers enabling greater throughput. FCP provides access to open systems SCSI FBA devices and the storage area network (SAN).

To support FCP, a new Channel Path Identifier (CHPID) type is available that will be defined and used on the FICON Express adapter. FICON on System z connects to a Symmetrix FICON (EF) director; FCP connects to a Symmetrix Fibre (FA) director.

Configuring EMC Symmetrix Arrays for Linux on System z

Applied Technology
Once you’ve chosen the connection protocol, you need to define the z/VM environment in the Input/Output Configuration Program (IOCP) deck. To add the definitions you must provide the Channel Path Identifier (CHPID), Control Unit Information (CNTLUNIT), and device address range (IODEVICE). Together these define the I/O devices under z/VM. An IOCP deck will be discussed in each section pertaining to FCP and FICON.

This paper will specifically focus on attaching a Symmetrix array to a System z running z/VM 5.4 using direct-attached ECKD and FBA devices.

### Linux on System z disk options

The basic connectivity choice of FICON or FCP CHPID drives the disk type options available in the environment. For FICON, the devices are ECKD; for FCP the devices are FBA. There are several ways to set up ECKD or FBA devices. Devices can be configured as dedicated or Minidisks.

- **Dedicated devices** to the Linux guest virtual machine either as ECKD or FBA. This paper will focus on these options.
- **Minidisks** — full or partial allocation of the z/VM I/O device— either ECKD or FBA, which are defined to the Linux guest virtual machine. FBA devices specifically use a z/VM edev device type. Discussion of the use of minidisks is beyond the scope of this paper.

There are various reasons why one method is chosen over the other. Some Linux environments use both device types on the Linux virtual machines. Extended CKD is very familiar to mainframe users and commonly used for the Linux operating system portion of the environment. FBA SAN devices are more familiar to open systems and used for user data. The Symmetrix environment supports all z/VM supported device types.

As there is flexibility when choosing disk types, they should be evaluated individually by company, policies, and application environment. Some disk types are easier to track and manage from z/VM, some from Linux, while other types require more or less storage provisioning. Whichever disk type is chosen, EMC recommends creating gatekeeper devices (small Symmetrix devices used for communication paths to the Symmetrix array) as a dedicated communication path to the Symmetrix array from a management host. This allows Symmetrix management software to retrieve configuration and status information without interfering with normal Symmetrix I/O.

### FCP connectivity example

Figure 1 shows a test environment that exemplifies how to connect FBA devices. The configuration consists of a Symmetrix array, a System z10 LPAR running z/VM 5.4, and two Linux guest virtual machines both using FCP devices. There are two Symmetrix FC directors of 1c:0 and 16c:0, each having their own unique WWPN and attached to a SAN switch. There are two System z channels defined as FCP, 8c and 8d, also attached to the SAN switch. These also have unique WWPNs. There are Symmetrix Logical Volumes presented to both directors to each channel. Together they comprise a SCSI FBA Linux environment.

This section discusses the steps to configure a SCSI FBA environment bringing LUNs online to a Linux guest virtual machine.
Symmetrix FCP requirements

When using FCP, the Symmetrix array has specific requirements for attaching to a z/VM and Linux environment. There are several front-end director (FA) options that must be set in the Symmetrix array for FCP. The director bit settings listed in Table 1 can be modified using EMC Solutions Enabler SYMCLI or Symmetrix Management Console (SMC), or can be set at the Symmetrix by a EMC Customer Engineer (CE).

Table 1. Symmetrix director bit settings

<table>
<thead>
<tr>
<th>Director bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>Point-to-Point</td>
</tr>
<tr>
<td>SPC2</td>
<td>SPC2 SCSI Primary</td>
</tr>
<tr>
<td>EAN</td>
<td>Enable Auto Negotiation</td>
</tr>
<tr>
<td>C</td>
<td>Common Serial Number</td>
</tr>
<tr>
<td>SC3</td>
<td>SCSI 3 Interface</td>
</tr>
<tr>
<td>UWN</td>
<td>Unique Worldwide Name</td>
</tr>
</tbody>
</table>

Note: A qualified storage administrator should make these changes in a controlled fashion. The Linux on IBM System z: RHEL 5.x and SLES 10.x Installation and Configuration Guide available on EMC Powerlink® discusses the requirements.
Adding FCP devices to Linux on System z

In order for the host to access storage it must be presented to it. The following are the basic steps to adding Symmetrix FBA devices to a Linux environment:

1. Set up hardware and make all physical connections between the Symmetrix array, System z, and the SAN switch.
2. Set up zoning on the switch to associate the appropriate components with the System z channel.
3. On the Symmetrix, define Symmetrix FBA LUNs, and map and mask to the appropriate front-end adapter port.
4. Add I/O definitions to z/VM through the IOCP deck.
5. Create z/VM virtual machine directory entries with assigned IODEVICE(s) in the directory.
6. Install Linux in the newly allocated Linux guest virtual machine.
7. Add an additional disk to the Linux virtual machine, if needed, via z/VM directory entry or the CP attach command (class B privilege required). For more information about privilege classes, see the “Privilege Classes” section in the z/VM: CP Commands and Utilities Reference on the IBM website.
8. Vary the Linux I/O device bus-ID online.
9. Associate the WWPN with the assigned I/O device bus-ID address.
10. Associate the LUN (SLV) with the appropriate Symmetrix FA WWPN.
11. Partition the Linux device /dev/sdX.
12. Add the Linux device (/dev/sdX) to Logical Volume Manager (LVM) and/or create a filesystem.

Building the Linux on System z FBA device relationship

All definitions that are created in this process are building a path relationship between the Symmetrix array, SAN, the System z LPAR, z/VM, and the guest virtual machine running Linux. Figure 2 illustrates the path relationship.

---

**Figure 2. Linux FBA device path relationship**

The following steps build the path from the Symmetrix array to the Linux guest virtual machine:
1. Create the Symmetrix FBA Logical Volumes. These will be mapped and masked out the designated director ports on the Symmetrix. During this process each SLV is assigned an internal device address and an external LUN address. In this example, the Linux host will use the external LUN to reference its device since it is a direct-attached z/VM device. Each FA director has a unique WWPN that is attached to a SAN switch. The SAN switch connects the Symmetrix array to the System z.

Additionally, the z/VM channel (System z FICON Express adapter loaded with FCP microcode) is also attached to the SAN switch, which has its own unique WWPN. The Symmetrix FA port (WWPN) is “zoned” together with the System z adapter (WWPN). During the definition process a z/VM physical System z adapter is assigned a CHPID as a type of FCP with its own unique WWPN. In the IOCP definition the z/VM CHPID has IODEVICEs associated to it, identified by a four-digit hexadecimal I/O device number.

2. Attach the z/VM IODEVICE to the Linux guest virtual machine as a real or virtual device address. This occurs either in the z/VM directory entry for the Linux virtual machine or through the z/VM CP attach command.

Once the I/O device is attached to the Linux guest virtual machine, Linux recognizes the attached I/O device address automatically by its own kernel hotplug routine. This is when the Linux device bus-ID is set up in the sysfs filesystem structure. The z/VM defined (dedicated or attached) I/O device and the Linux device bus-ID are the same four digits in this example but can be different.

Note: The real and virtual addresses can be different. When a z/VM defined I/O device is defined to the Linux guest virtual machine, through the z/VM attach or dedicate command, you must specify a virtual address or leave it as the real, z/VM address.

As this is a FCP environment, the device bus-ID has a WWPN associated to it. It will be used to define the FCP environment, further building on the relationship path between the Symmetrix and Linux guest virtual machine.

3. Associate a LUN address, which is also the SLV external LUN assigned during SLV creation within the Symmetrix, to the WWPN of the Symmetrix FA. When the LUN is varied online, the LUN is associated to a Linux device name or node such as /dev/sdX.

4. Partition the Linux device name and then add it to the LVM environment or to the Linux filesystem, or to both.

Now that we know the steps and the path relationship between the components, we will build the environment shown in Figure 2. These steps are discussed in the “z/VM IOCP example” section next through the “Configuring FCP devices on Linux on System z” section on page 13.

**z/VM IOCP example**

Once the physical connections, zoning, and Symmetrix FBA devices are created, the Symmetrix Logical Volumes (SLVs) must be defined to z/VM. A z/VM systems programmer performs this task in the IOCP deck.

The following is a sample IOCP deck from the z/VM test environment. Notice the TYPE is specified as FCP for FBA devices. There are two channels with 250 devices each providing dual paths to each device when set up properly.

```
CHPID PATH=(CSS(0),8C),SHARED,PARTITION=((0),(V109)),         *
PCHID=392,TYPE=FCP
CHPID PATH=(CSS(0),8D),SHARED,PARTITION=((0),(V109)),         *
PCHID=503,TYPE=FCP
CNTLUNIT CUNUMBR=0600,PATH=((CSS(0),8C)),UNIT=FCP
IODEVICE ADDRESS=(6500,25),CUNUMBR=(0600),UNIT=FCP
CNTLUNIT CUNUMBR=0640,PATH=((CSS(0),8D)),UNIT=FCP
IODEVICE ADDRESS=(6500,25),CUNUMBR=(0640),UNIT=FCP
```

Configuring EMC Symmetrix Arrays
for Linux on System z
Applied Technology  9
Linux on System z z/VM directory entry example

The next step is to define a virtual machine by setting up a z/VM directory entry. It can be performed manually or through z/VM’s facility DIRMAINT, where you assign the virtual machine certain properties. The example below shows the directory entry for the environment used in this paper. It has the following basic properties:

- USER name, in this case LN142197
- Password of the virtual machine, in this case LINUX
- Minimum and maximum virtual machine memory, in this case 512M and 1024M, respectively
- Privilege class

Note: In Linux virtual machines it is recommended to run as a class G user privilege to protect the z/VM environment. This privilege class lets users run only commands that can affect their own virtual machine.

There is a default profile setup for this user, LNXDFLT. This is where the z/VM systems programmer can specify certain common characteristics for a virtual machine, such as setting up an Integrated Facility for Linux (IFL).

Most of the entries in a directory are the definition of their I/O devices or in this case the FBA SLVs. Notice the DEDICATE statements in the following example. The SLVs 6580 and 6680 defined in the IOC deck in the “z/VM IOC example” section are dedicated solely for the use of this virtual machine. The devices found at these I/O addresses are the entire SLV that was defined on the Symmetrix array, not a z/VM minidisk. When you log in to z/VM and instantiate the virtual machine the devices in the z/VM directory will be attached, or dedicated to your virtual machine for its sole use.

```
USER LN142197 LINUX 512M 1024M G
INCLUDE LNXDFLT
OPTION QUICKD
DEDICATE 0191 6C03
DEDICATE 6580 6580
DEDICATE 6680 6680
```

Once the z/VM directory is created, log in as the specified user, verify the devices are attached to your Linux virtual machine, and install the Linux environment.

Note: When you are logged in to z/VM, this is considered the VM console for your Linux virtual machine. Occasionally, messages may appear, such as when adding new devices to your Linux virtual machine environment.

After successfully installing Linux, you can now log in to your Linux guest through telnet or ssh. This will be considered your Linux session.

**Querying FCP devices**

After logging in to the Linux virtual machine, look at the FCP devices at the z/VM level to determine which devices are defined to its virtual machine. You can accomplish this two ways: One is from the virtual machine, and the other is if Linux is installed. You can do the latter through the vmcp utility that comes with current Linux environments. The manpage for vmcp states “vmcp allows Linux users to send commands to the control program of z/VM.”

```
Note: Whether you execute commands from Linux or from the VM console, you are communicating with z/VM to get the FCP device information.
```

Configuring EMC Symmetrix Arrays for Linux on System z
Applied Technology
Before you can execute vmcp from Linux, the kernel module must be loaded. Use the following command to verify that it is loaded:

```
# lsmod|grep vmcp
vmcp               24584   0
```

If the kernel module is not loaded, enter the following command to load the vmcp module:

```
# modprobe -v vmcp
insmod /lib/modules/2.6.16.60-0.21-default/kernel/drivers/s390/char/vmcp.ko
```

After loading the vmcp kernel module, execute the `lsmod` command again to verify its presence.

Once vmcp is available on your Linux environment, you can execute z/VM CP commands. Use the following command to view all the FCP devices allocated to the Linux virtual machine:

```
# vmcp q fcp
```

In the following example the FCP devices are already being used as indicated by the IOCNT fields.

```
[root@ln142197 lib]# vmcp q fcp|more
FCP  6580 ON FCP   6580 CHPID 8C SUBCHANNEL = 000D
      TOKEN      = 00000001FAF38B00
6580 DEVTYPE FCP         CHPID 8C FCP
6580 QDIO ACTIVE        QIOASSIST ACTIVE        QEBSM
5680
6580 INP + 01 IOCNT = 00000802  ADP = 128 PROG = 000 UNAVAIL = 000
      TOKEN      = 00000001EDDE580
6580 OUT + 01 IOCNT = 00000911  ADP = 000 PROG = 128 UNAVAIL = 000
5680
5680 INP + 01 IOCNT = 00000102  ADP = 128 PROG = 000 UNAVAIL = 000
      TOKEN      = 00000001EDDE580
6580 OUT + 01 IOCNT = 00000111  ADP = 018 PROG = 110 UNAVAIL = 110
5680
```

If Linux is not already installed, but you are logged in to a Linux VM console session, you can still examine the FCP devices as listed in this next example. From the z/VM prompt, enter: `q fcp`. This will display devices allocated, or attached, to the virtual machine you are logged in to.

```
q fcp

FCP  6580 ON FCP   6580 CHPID 8C SUBCHANNEL = 0010
      TOKEN      = 00000001EDDE580
6580 DEVTYPE FCP         CHPID 8C FCP
6580 QDIO ACTIVE        QIOASSIST ACTIVE        QEBSM
6580
6580 INP + 01 IOCNT = 00000002  ADP = 128 PROG = 000 UNAVAIL = 000
      TOKEN      = 000000001F000000
6580 OUT + 01 IOCNT = 00000011  ADP = 000 PROG = 018 UNAVAIL = 110
6580
```

**Bringing FCP devices online**

At this point, the guest virtual machine directory is created, FCP devices are defined to it at the z/VM level through the z/VM directory, and Linux is successfully installed and started.

You can now bring the FCP devices online to Linux. Bringing devices online is more of a mainframe notion than an open systems Linux procedure. However, before you can use the devices in a Linux environment, the devices must first be brought online. This can be accomplished in two ways.

- **Method 1** — `echo 1` into the file, `online`. This file exists within the devices’ filesystem structure as shown in the following example:

  ```
  #:/sys/bus/ccw/devices/0.0.6580 # cat /sys/bus/ccw/devices/0.0.6580/online
  0
  #:/sys/bus/ccw/devices/0.0.6580 # echo 1 > /sys/bus/ccw/devices/0.0.6580/online
  ```
The Linux `cat` command shows what is within the file. The file `online` has a 0 in it which means the device is offline or not available for use. After a 1 is placed into the file with the `echo` command, the device is brought online. However, check the file again with `cat`, to make sure the device was successfully brought online. If Linux was not able to successfully bring the device online you will still see a 0 in the file.

- **Method 2** — Bring devices online using the `chccwdev` script. In this example, you pass the parameter `--online` and the device number, 0.0.6580. The device number used should be the same that resulted from the query of FCP devices previously executed. If the command is successful, the following message displays indicating that it is setting the device online:

```
ln142197:/sys/bus/ccw/devices # chccwdev --online 0.0.6580
Setting device 0.0.6580 online
```

When bringing a device online to Linux you may also see messages similar to the following appear on the virtual machine console for your Linux guest. This shows establishment of communication between z/VM, Linux, and the FCP device (SLV) that exists on the Symmetrix array.

```
scsi4 : zfcp
zfcp: The adapter 0.0.6580 reported the following characteristics:
WWNN 0x5005076400c6cefe, WWPN 0x5005076401a22154, S_ID 0x007b7813,
adapter version 0x4, LIC version 0x70b, FC link speed 2 Gb/s
zfcp: Switched fabric fibrechannel network detected at adapter 0.0.6580.
```

Note: This message is from the z/VM adapter standpoint and not the Symmetrix array. The WWNN and WWPN are the z/VM adapter’s information, not the Symmetrix WWPN. The FC link speed is the speed of the z/VM adapter, not the speed the Symmetrix array has available to it.

**Determining the WWPN – FCP**

Once the FCP devices are online to Linux, you will need to get the WWPN of the Symmetrix front-end director for the next step. Here are two ways to determine the WWPN for the Symmetrix FA.

- **Via the SAN switch.** In order to get the WWPN information you will either need authority to access the switch or get a print screen of the zone information containing the Symmetrix FA port. A print screen also lets you verify what is in the zone.
- **Via EMC Solutions Enabler,** from another host where Solutions Enabler is installed, execute the `symcfg` command with the following parameters.

Once you have this information, the WWPN information is used during addition of the FCP LUNs. If you’ve installed the base operating system to FCP SAN devices it’s needed to IPL Linux. When booting from a FCP device use the `set loaddev` command.

To determine the WWPN for the FA, execute the following Solutions Enabler command `symcfg`. The following is partial output of the symcfg command showing WWN information.

```
symcfg -fa 1c list -sid 571 -v -P 0
where -fa is the director you are using for FCP
where -sid is the unique Symmetrix serial number identifier
where -P is the port being used on the director
```

```
Director Symbolic Number : 01C
Director Numeric Number : 33
Director Slot Number : 1
```

Configuring EMC Symmetrix Arrays
for Linux on System z
Applied Technology 12
Configuring FCP devices on Linux on System z

Use the WWPN for the Symmetrix FA to execute the next step in adding FCP devices to your Linux environment. In the following command we’re associating the WWPN 0x5006048ad5f066c0 to a specific Linux device bus-ID, in this case 0.0.6580:

1. Add the Symmetrix system WWPN for the FCP device bus-ID:
   
   ```bash
   echo 0x5006048ad5f066c0 > /sys/bus/ccw/drivers/zfcp/0.0.6580/port_add
   ```

2. Associate a specific LUN with the WWPN and thus the Linux device bus-ID, 0.0.6580.

   Add the LUN using the WWPN and LUN number. The LUN number is the first four bytes that are echoed into the file. Also, this is the external LUN assigned and mapped to the SLV; it is not the internal SLV identifier. Once this is done, the path from the SLV to the Linux LUN shown in Figure 3 is built.

   ```bash
   echo 0x0080000000000000 >/sys/bus/ccw/drivers/zfcp/0.0.6580/0x5006048ad5f066c0/unit_add
   ```

   ![Figure 3. Path from the SLV to the Linux LUN](image)

VM console message

When associating the Linux device bus-ID with the Symmetrix FA port, the following message for its WWPN and specific LUN displays on the virtual machine console. This message shows the Linux device name assignment. In this instance, `sde`, also known as the device name, is associated with the Linux device.
bus-ID, 0.0.6580, and the defined WWPN. For the next step of partitioning a new SCSI disk, you need to know the Linux device name to run the `fdisk` command.

```
Vendor: EMC       Model: SYMMETRIX       Rev: 5773
Type: Direct-Access       ANSI SCSI revision: 04
SCSI device sde: 1839360 512-byte hdwr sectors (942 MB)
sde: Write Protect is off
SCSI device sde: drive cache: write through
SCSI device sde: 1839360 512-byte hdwr sectors (942 MB)
sde: Write Protect is off
SCSI device sde: drive cache: write through
sde: unknown partition table
sd 0:0:0:128: Attached scsi disk sde
sd 0:0:0:128: Attached scsi generic sg4 type 0
```

Note: Receiving this message on the Linux virtual machine console confirms communication from Linux through z/VM to the Symmetrix array. Notice the vendor and SCSI device attributes (SCSI equates to FBA) that are reported. If these attributes are not what you think they should be, double-check the procedures to verify the correct WWPN and LUN were used when adding the device.

If the message does not display on the virtual machine’s console, execute Linux commands to list SCSI devices. There are many ways to get this information on a Linux guest.

On SUSE, you can list SCSI devices using the `lsscsi` command. Notice in the output below it shows the Linux device bus-ID, 0.0.6580. The 6580 equates to the attached z/VM I/O device number defined in the IOCP deck, the Linux device name (/dev/sde), the disk attributes (EMC, Symmetrix, and 5773, which is the EMC Enginuity™ version), and so on.

```
ln142197:/sys/bus/ccw/devices/0.0.6580 # lsscsi -v
---
[0:0:0:128]    disk    EMC      SYMMETRIX        5773  /dev/sde
dir: /sys/bus/scsi/devices/0:0:0:128
[/sys/devices/css0/0.0.0001f/0.0.6580/host0/rport-0:0-0/target0:0:0/0:0:0:128]
```

**Partition a SCSI device via fdisk**

To get a Linux device ready for use for LVM or to directly place a filesystem on the disk, you must partition the device. You will need the Linux device name and use the `fdisk` command. In the following, one partition will be placed on device /dev/sde.

```
# fdisk /dev/sde
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel. Changes will remain in memory only, until you decide to write them. After that, of course, the previous content won’t be recoverable.

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)

Command (m for help): p
Partition number (1-4): 1
First cylinder (1-1023, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1023, default 1023):
Using default value 1023
```

Configuring EMC Symmetrix Arrays
for Linux on System z
Applied Technology 14
Creating a filesystem

After partitioning the disk, you can either add it to LVM or create a filesystem on the device. The following shows how to create a filesystem on the device using the primary partition we just created on /dev/sde.

```
[root@ln142197 0.0.6580]# mke2fs -j /dev/sde1
mke2fs 1.39 (29-May-2006)
.............
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.
```

ECKD connectivity example

Setting up ECKD dedicated DASD requires fewer steps than doing so in an FCP environment because you don’t have a SAN environment involved. For this reason some existing mainframe shops prefer ECKD over FBA devices.

This section we will examine the steps to bring these devices online to a Linux guest virtual machine.

**ECKD test environment**

Figure 4 illustrates a test environment that shows the overall setup for CKD devices. The environment contains a Symmetrix array, a System z10 LPAR running z/VM 5.4, and two Linux guest virtual machines both using CKD devices. The Symmetrix array provides the FICON director ports 2c:0 and 15c:0. There are two channels defined as FC (FICON), 8e and 8f, which are directly attached to the Symmetrix front-end director ports.
Adding ECKD devices to Linux on System z

The following are the basic steps for adding ECKD devices directly to a guest virtual machine:
1. Set up hardware and make all physical connections.
2. Add I/O definitions to z/VM via the IOCP deck.
3. Create z/VM directory entries with the assigned disk dedicated to the virtual machine.
4. Log in to the new virtual machine user.
5. Install Linux in the new Linux guest virtual machine.
6. Optionally, add an additional disk to the Linux virtual machine via directory entry or the CP attach command.
7. Vary Linux devices online.
10. Add to LVM or create a filesystem directly on the partitioned device, or both.

Building the Linux on System z ECKD device relationship

When setting up any environment there is always a relationship between the components. Figure 5 shows the relationship that we are building with the commands documented herein. All definitions created in this process are building a path relationship between the Symmetrix array, the System z LPAR, z/VM, and the guest virtual machine running Linux.
1) **Symmetrix – ECKD device created and mapped:**
   a) The physical disks are grouped to create Symmetrix Logical Volumes (SLV). At the same time an internal device address is assigned. The newly created SLV is assigned an address required for host access.
   b) Map the SLV to the front-end FICON director port (EF), also known as the director, where the FICON port is attached directly to mainframe, or optionally to a FICON switch, which is very common. The optional switch connects the Symmetrix system to the mainframe.

2) **z/VM – Define and identify the z/VM I/O devices.** On the mainframe, a System z FICON Express adapter is defined as a CHPID TYPE FC to set up FICON connectivity. The I/O environment is defined to z/VM by listing the CHPID and its IODEVICES in the IOCP deck.
   
   Once the z/VM I/O device is defined and identified, attach them to the Linux guest virtual machine as real or virtual device addresses. Use the z/VM CP attach command or define the DASD in the guest virtual machine’s directory, which is discussed next.

3) **Linux –** When Linux is assigned a z/VM IODEVICE the attached or dedicated device number is recognized by Linux through the kernel hotplug routine. This new device is known as a Linux device bus-ID. By running commands, documented herein, the Linux bus-ID, or more commonly known as a LUN in open systems, is associated to a Linux device name, /dev/dasdX, as opposed to /dev/sdX for FBA devices.

   After associating the Linux bus-ID to the Linux device name, format the device using the dasdfmt utility. Finally, either add it to LVM, or create a Linux filesystem, or do both on the DASD device.

---

**Figure 5. Linux ECKD device path relationship**
**IOCDS example for Symmetrix – CKD**

After you have the physical connections established and the Symmetrix CKD devices are created, define the Symmetrix Logical Volumes (SLVs) to z/VM. This work is done by the z/VM systems programmer. The following is a sample entry from the z/VM test environment. Notice the TYPE is specified as FC for CKD devices versus a TYPE of FCP for FBA devices. There are 250 devices assigned two channels providing dual channel paths to each device.

```
CHPID PATH=(CSS(0),8E),SHARED,                                              *
                  V109)),PCHID=593,TYPE=FC *
CHPID PATH=(CSS(0),8F),SHARED,                                              *
                  V109)),PCHID=5E0,TYPE=FC *
CNTLUNIT CUNUMBR=0680,PATH=((CSS(0),8E,8F)),                               *
    UNITADD=((00,256)),CUADD=0,UNIT=2105 IODEVICE ADDRESS=(1200,250),CUNUMBR=(0680),STADET=Y,UNIT=3390
```

**Linux on System z VM directory entry example**

As previously mentioned in the FBA example, a VM directory needs to be set up with the proper devices. Devices can either be added to an existing virtual machine directory, defined in a new VM directory entry, or CP attached to a virtual machine. You can perform this step manually or through z/VM’s facility, DIRMAINT, and assign the virtual machine certain properties. The following is the directory entry for the CKD environment used herein, which has some basic properties:

- **USER** name, in this case LN165017
- **Initial password** of LINUX
- **Minimum and maximum virtual machine memory**, 512M and 1024M, respectively
- **Privilege class** (G). As previously mentioned, in Linux virtual machines, it is recommended running as a class G user privilege to protect your z/VM environment. This privilege class lets users run only commands that can affect their own virtual machine.

```
USER LN165017 LINUX 512M 1024M G
INCLUDE LNXDFLT
OPTION QUICKD
```

**Note:** Notice that a default profile is set up for this user, LNXDFLT. This is where the z/VM systems programmer can specify certain common characteristics for a virtual machine – for example, the setup and use of an IFL.

Most of the entries in a directory are the definition of their I/O devices or in this case the CKD SLVs. DEDICATE statements are shown below. The example also shows the SLVs defined in the IOCP deck previously discussed, 12ca-12cb and 1250-1251, are dedicated solely for the use of this virtual machine. The devices found at these I/O addresses are the entire SLVs that were defined on the Symmetrix array, not z/VM minidisks. When you log in to z/VM and instantiate the virtual machine, the devices in your z/VM directory will be attached or dedicated to your virtual machine for its sole use.

```
USER LN165017 LINUX 512M 1024M G
INCLUDE LNXDFLT
OPTION QUICKD
DEDICATE 0191 1203
DEDICATE 0100 12CA
DEDICATE 0101 12CB
DEDICATE 1250 1250
DEDICATE 1251 1251
```

Once the VM directory is created, you can log in as the specified user, verify the devices are attached to your Linux virtual machine, and install the Linux environment. When you are logged in to the Linux virtual machine this is considered the VM console for your Linux virtual machine. Occasionally, certain messages may appear (see the “VM console message” section); for instance, when adding new devices to your Linux virtual machine environment.
After successfully installing Linux, you can now log in to your Linux guest through telnet or ssh. This will be considered your Linux session.

**Viewing ECKD DASD on z/VM**

Once you have your virtual machine set up, look at your CKD devices at the z/VM level to see which devices are defined to the Linux virtual machine. Do this before you begin the Linux installation to verify all the required DASDs are defined. This can be done from your Linux virtual machine console session by querying z/VM CP with the `Q DASD` command. You will get information about all your virtual machine DASD devices, but look for the DASD statements that pertain to your requested Linux Symmetrix storage and defined in the z/VM IOPC deck.

Notice there are other 3390 devices that you did not request. These are part of the virtual machine’s devices to support the z/VM CMS environment and its functions. These also exist in the FCP environment’s directory entry.

```plaintext
Q DASD
DASD 0100 ON DASD 12CA R/W 0X0100 SUBCHANNEL = 000C
DASD 0101 ON DASD 12CB R/W 0X0101 SUBCHANNEL = 000D
DASD 0190 3390 $Z11RS R/O 107 CYL ON DASD 0288 SUBCHANNEL = 0007
DASD 0191 3390 $Z11U1 R/O 100 CYL ON DASD 028D SUBCHANNEL = 000A
DASD 019D 3390 $Z11RS R/O 146 CYL ON DASD 0288 SUBCHANNEL = 0008
DASD 019E 3390 $Z11RS R/O 250 CYL ON DASD 0288 SUBCHANNEL = 0009
DASD 0592 3390 $Z11RS R/O 70 CYL ON DASD 0288 SUBCHANNEL = 000B
DASD 1250 ON DASD 1250 R/W SM1250 SUBCHANNEL = 000F
DASD 1251 ON DASD 1251 R/W 0X6C51 SUBCHANNEL = 000F
DASD 1252 ON DASD 1252 R/W 0X6C52 SUBCHANNEL = 0010
DASD 1253 ON DASD 1253 R/W 0X6C53 SUBCHANNEL = 0011
DASD 1254 ON DASD 1254 R/W 0X6C54 SUBCHANNEL = 0012
DASD 1255 ON DASD 1255 R/W SYM13D SUBCHANNEL = 0013
DASD 1256 ON DASD 1256 R/W SYM13E SUBCHANNEL = 0014
DASD 1257 ON DASD 1257 R/W SYM13F SUBCHANNEL = 0015
DASD 1258 ON DASD 1258 R/W LNX10X SUBCHANNEL = 0016
DASD 1259 ON DASD 1259 R/W SYM141 SUBCHANNEL = 0017
```

If Linux is installed, run the same command using `vmcp` that comes with current Linux environments. If it is not active when you execute `vmcp`, load the `vmcp` module as discussed in the “Querying FCP devices” section.

In the following output, some of the devices are attached, or dedicated, as virtual addresses as shown with devices 0100 and 0101. Their real z/VM assigned addresses are 12CA and 12CB accordingly. The other DASDs shown are attached as their z/VM or real addresses.

```plaintext
ln165017:~ # vmcp q dasd
DASD 0100 ON DASD 12CA R/W 0X0100 SUBCHANNEL = 000C
DASD 0101 ON DASD 12CB R/W 0X0101 SUBCHANNEL = 000D
DASD 0190 3390 $Z11RS R/O 107 CYL ON DASD 0288 SUBCHANNEL = 0007
DASD 0191 3390 $Z11U1 R/O 100 CYL ON DASD 028D SUBCHANNEL = 000A
DASD 019D 3390 $Z11RS R/O 146 CYL ON DASD 0288 SUBCHANNEL = 0008
DASD 019E 3390 $Z11RS R/O 250 CYL ON DASD 0288 SUBCHANNEL = 0009
DASD 0592 3390 $Z11RS R/O 70 CYL ON DASD 0288 SUBCHANNEL = 000B
DASD 1250 ON DASD 1250 R/W SM1250 SUBCHANNEL = 000F
DASD 1251 ON DASD 1251 R/W 0X6C51 SUBCHANNEL = 000F
DASD 1252 ON DASD 1252 R/W 0X6C52 SUBCHANNEL = 0010
DASD 1253 ON DASD 1253 R/W 0X6C53 SUBCHANNEL = 0011
DASD 1254 ON DASD 1254 R/W 0X6C54 SUBCHANNEL = 0012
DASD 1255 ON DASD 1255 R/W SYM13D SUBCHANNEL = 0013
DASD 1256 ON DASD 1256 R/W SYM13E SUBCHANNEL = 0014
DASD 1257 ON DASD 1257 R/W SYM13F SUBCHANNEL = 0015
DASD 1258 ON DASD 1258 R/W LNX10X SUBCHANNEL = 0016
DASD 1259 ON DASD 1259 R/W SYM141 SUBCHANNEL = 0017
```
**Configuring ECKD devices to Linux**

If all disks devices are not present during the initial setup or IPL of Linux, the disks can easily be added later. The hotplug routine adds the devices to Linux when they're attached to a running Linux virtual machine.

To view all DASDs, online and offline, use the following command:

```
ln165017:~ # lsdasd -a
0.0.1250(none) : offline
0.0.1251(none) : offline
0.0.1252(none) : offline
0.0.1253(none) : offline
0.0.1254(none) : offline
0.0.1255(none) : offline
0.0.1256(none) : offline
0.0.1257(none) : offline
0.0.1258(none) : offline
0.0.1259(none) : offline
0.0.0100(ECKD) at ( 94:  0) is dasda      : active at blocksize 4096, 1803060 blocks, 7043 MB
0.0.0101(ECKD) at ( 94:  4) is dasdb      : active at blocksize 4096, 1803060 blocks, 7043 MB
```

**Bringing the ECKD device online**

Before you can use the ECKD device, you need to bring the device online to Linux. This differs from x86 Linux environments, but is familiar to mainframe systems programmers. The procedure is performed using the Linux `chccwdev` script. This script contains a group of commands that modify attributes of channel attached devices. Bringing devices online or offline is one function that `chccwdev` offers.

```
ln165017:~ # chccwdev --online 0.0.1250
Setting device 0.0.1250 online
Done
```

If the device is not available or `chccwdev` cannot successfully bring it online and an error message appears. Use the `lsasd -a` command to check the device status or examine the status of the device using the `cat` command. Navigate to the device directory and `cat` the file called `online` as shown. If the device is online, the content of the file is “1” as the following example shows. If the content is “0”, the device is offline.

```
ln165017:/sys/bus/ccw/devices/0.0.1250 # cat /sys/bus/ccw/devices/0.0.1250/online
1
```

**Format ECKD devices**

After the ECKD device is successfully brought online to Linux, format the device prior to its use to prepare it for data usage. To format a device, provide the Linux device name or node, /dev/dasdX, where X is a letter(s) assigned by Linux after the device is brought online. This assignment is in the device’s sysfs directory structure. You can use the `lsasd` command.

```
ln165017:/sys/bus/ccw/devices/0.0.1250 # ls
alias  bus  devtype  eer_enabled  readonly  use_diag
availability  cmb_enable  discipline  modalias  uevent  vendor
block:dasdc  cutope  driver  online  uid
```

---

**Configuring EMC Symmetrix Arrays**

for Linux on System z

Applied Technology  20
In the following example, the command has the following attributes:

- `-b 4096` blocksize 4096
- `-d cdl` compatible disk layout which is the default. This prepares the device so it can also be accessed by z/OS.
- `-l sm1250` assigns a label which is optional
- `-v` verbose
- `-f` complete filesystem path name to the device node

```
ln165017:/sys/bus/ccw/devices/0.0.1250 # dasdfmt -b 4096 -d cdl -l sm1250 -v -f
/dev/dasdc
```

Retrieving disk geometry...
Drive Geometry: 1113 Cylinders * 15 Heads = 16695 Tracks

I am going to format the device /dev/dasdc in the following way:

- Device number of device: 0x1250
- Labelling device: yes
- Disk label: VOL1
- Disk identifier: SM1250
- Extent start (trk no): 0
- Extent end (trk no): 16694
- Compatible Disk Layout: yes
- Blocksize: 4096

--- ATTENTION! ---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
Detaching the device...
Invalidating first track...
Formatting tracks complete...
Re-accessing the device...
Finished formatting the device.
Retrieving dasd information... ok
Writing empty bootstrap... ok
Writing label... ok
Writing VTOC... ok
Rereading the partition table... ok

After formatting the device, partition it for use with fdasd. Partitioning a device gets it ready for LVM or creating a filesystem on the device:

```
ln165017:/dev/.udev/db # fdasd /dev/dasdc
```

reading volume label ..: VOL1
reading vtoc ..........: ok

Command action
m   print this menu
p   print the partition table
n   add a new partition
d   delete a partition
v   change volume serial
t   change partition type
r   re-create VTOC and delete all partitions
u   re-create VTOC re-using existing partition sizes
s   show mapping (partition number - data set name)
q   quit without saving changes
w   write table to disk and exit

Command (m for help): n
First track (1 track = 48 KByte) ([2]-16694):
Using default value 2
Last track or +size[c|k|M] (2-[16694]):
Using default value 16694

Command (m for help): w
writing VTOC...
rereading partition table...

ln165017:/dev/.udev/db # fdasd /dev/dasdc
reading volume label ..: VOL1
reading vtoc ..........: ok

Command action
  m print this menu
  p print the partition table
  n add a new partition
d delete a partition
  v change volume serial
t change partition type
  r re-create VTOC and delete all partitions
  u re-create VTOC re-using existing partition sizes
  s show mapping (partition number - data set name)
  q quit without saving changes
  w write table to disk and exit

Command (m for help): p

Disk /dev/dasdc:
cylinders ............: 1113
tracks per cylinder ..: 15
blocks per track ......: 12
bytes per block ......: 4096
volume label ..........: VOL1
volume serial ..........: 0X1250
max partitions ........: 3

--------------------------------------- tracks ---------------------------------------
Device      start      end   length   Id  System
/dev/dasdc1          2    16694    16693    1  Linux native

Command (m for help): quit

DASDFMT under z/VM

DASDFMT is a Linux utility that prepares a CKD device for data use. For Symmetrix arrays using devices configured with RAID 5 or RAID 6 protection, the following APARs must be applied to the z/VM and Linux on System z environments. The format can run without these fixes, but performance will be significantly improved when these are applied. For further information, refer to EMC Knowledgebase cases emc207283 and emc207585.

Apply the following four fixes — two IBM APARs are for z/VM; two fixes are at the Linux on System z level.

- For the z/VM environment, apply:
  - For CMS — VM64602: CMS FORMAT OF A MINIDISK IS MUCH SLOWER THAN CPFMTXA
    CMS FORMAT of a minidisk or a full pack minidisk is slower than using CPFMTXA. This is due to the fact that the CCW chain created by CMS FORMAT does not set define extent byte 7 bit 5.

  - For CP — VM64603: CCW TRANSLATION CHANGE FOR CMS FORMAT
    VM's CCW fast translation fails to allow guest I/O to a minidisk to use a performance feature related to writing out an entire track.
For the Linux on System z environment, the fixes are applied in two parts: One is at the kernel level; and the other is optional applied to the dasdfmt command. The dasdfmt command is considered optional since it provides the option to disable the feature in the dasdfmt command.

Note: When using RAID 5 or RAID 6 protection, EMC recommends applying all fixes to your environment.

- For Red Hat Red Hat Enterprise Linux 5.2, see Bugzilla 486432
  For RHEL 5.3, see Bugzilla 486431
- For Novell SUSE 10 SP2, see Bugzilla 450989
  For SUSE 11, see Bugzilla 477816

Multipathing
Multipathing is available at the System z level or implemented at the Linux level depending on your configuration. No matter which option you choose, other than the normal high availability planning, nothing is required for Symmetrix Logical Volumes. You will need to ensure that the SLVs are presented to two different director ports to the mainframe and to two different channels so there are two unique paths to each device.

If you have a SAN and dual fabrics, it may be prudent to have each path to go through a different fabric and switch to ensure the highest availability.

If you use:
- DASD devices via ESCON or FICON, multipathing is available at the System z level. This is transparent at the Linux virtual machine level. Setup is inherent in the IOCP deck definition of the ECKD devices by utilizing and defining two channels at the System z level.
- SCSI disk via FCP attached devices, multipathing is handled at the Linux level. Support is available in the Linux v2.6.x kernel multipathing solution, DM-MPIO. DM-MPIO requires packages device-mapper, udev, hotplug, and device-mapper-multipath. You should follow the standard Linux multipathing instructions.

Note: The following guide available on EMC Powerlink.com contains greater detail: Native Multipath Failover Based on DM-MPIO for v2.6.x Linux Kernel and EMC Storage Arrays Configuration Guide. This guide covers Red Hat Enterprise Linux 4 and 5, SuSE Linux Enterprise Server 9 and 10, Oracle Enterprise Linux 4 and 5, and Asianux 2.0 and 3.0.
EMC Solutions Enabler for Linux on System z

Solutions Enabler is available to run on Linux SUSE Linux Enterprise Server (SLES 10) environments in addition to other open system hosts such as Windows, and UNIX environments. In the SUSE environment Solutions Enabler can be used for configuration, managing, and monitoring of a Symmetrix storage system.

EMC Solutions Enabler introduction

Symmetrix Command Line Interface (SYMCLI) provides a host with a comprehensive command set for managing a Symmetrix storage environment. It is invoked from the Linux command line. You can use the commands in scripts that may provide further integration with OS and application.

The Solutions Enabler current support environment is Novell SuSE Linux Enterprise 10, including Service Pack 1 and 2. Solutions Enabler installation for Linux is via a rpm. During this process, documented in the EMC Solutions Enabler Installation Guide, a kernel module is installed via the insmod command. You need to choose the correct module based upon your SUSE version. Please see the EMC Solutions Enabler Installation Guide for further installation information.

Solutions Enabler ECKD requirements

If SUSE is running under z/VM and using ECKD devices, ECKD gatekeeper devices must be defined as “unsupported” DASD via the z/VM rdev command. This allows the software to properly communicate to the Symmetrix devices without interfering with application I/O. The following line shows an example of the rdev command. The rdev devices should be set up before the devices are assigned to the Linux guest virtual machine. Once the rdev device is set up you must dedicate the device to the guest virtual machine.

set rdev XXXX type unsupported devclass dasd dps yes reserve_release yes

The following is an example of defining a rdev for ECKD DASD device 1240. Here are the steps which are exhibited.

1. Query to show the device details of DASD 1240.
2. Vary the device offline in order to create the rdev device.
3. Create the rdev device.
4. Query to show the device details of DASD 1240. Notice the “Invalid device type”.
5. Attach or dedicate the device to the Linux guest virtual machine.

q dasd details 1240

1240  CUTYPE = 2107-E8, DEVTYPE = 3390-02, VOLSER = SYM128, CYLS = 1113
      CACHE DETAILS: CACHE NVS CFW DFW PINNED CONCOPY
         -SUBSYSTEM Y Y Y - N N
         -DEVICE Y - - Y N N
      DEVICE DETAILS: CCA = 40, DDC = --
      DUPLEX DETAILS: --
      CU DETAILS: SSID = 0200, CUNUM = 1080
Ready; T=0.01/0.01 18:03:49

vary off 1240
1240 varied offline
1 device(s) specified; 1 device(s) successfully varied offline
Ready; T=0.01/0.01 18:03:55

Set RDEVice 1240 Type UNSUPported DEVClass DASD DPS Yes RESERVE_Release Yes
HCPZRP6722I Characteristics of device 1240 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) created
Ready; T=0.01/0.01 18:04:13

vary on 1240
1240 varied online
1 device(s) specified; 1 device(s) successfully varied online
Ready; T=0.01/0.01 18:04:25

q dasd details 1240
HCPQDE006E Invalid device type - 1240
If you want to clear the unsupported rdev DASD definition:
1. Query the device. It must first be detached from any virtual machine and FREE when queried.
2. Vary the device offline to clear the rdev definition.
3. Clear the rdev definition.
4. Vary the device, 1240 online.
5. Query to show the device details of DASD 1240.

```plaintext
q 1240
DEV 1240 FREE

vary off 1240
1240 varied offline
1 device(s) specified; 1 device(s) successfully varied offline
Ready; T=0.01/0.01 18:04:42

set rdev 1240 clear
HCPZRP6722I Characteristics of device 1240 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created

q 1240
DEV 1240 OFFLINE
Ready; T=0.01/0.01 18:04:54

vary on 1240
1240 varied online
1 device(s) specified; 1 device(s) successfully varied online
Ready; T=0.01/0.01 18:05:02

q dasd details 1240
1240 CUTYPE = 2107-EB, DEVTYP = 3390-02, VOLSER = SYM128, CYLS = 1113
CACHE DETAILS: CACHE NVS CFW DFW PINNED CONCOPY
   -SUBSYSTEM Y Y Y - N N
   -DEVICE Y - - Y N N
DEVICE DETAILS: CCA = 40, DDC = --
DUPLEX DETAILS: --
CU DETAILS: SSID = 0200, CUNUM = 1080

Ready; T=0.01/0.01 18:05:02
```

Once Solutions Enabler is fully installed and the devices properly defined, the devices will be seen as type, CGK as shown:

```plaintext
ln142196:~ # syminq
Device             Product                Device
------------------  ---------------------------------  
Name     Type   Vendor    ID           Rev  Ser Num      Cap (KB)
------------------  ---------------------------------  
/dev/dasdc CGK    EMC     SYMMETRIX    5773 7100138340       5760
/dev/dasdd CGK    EMC     SYMMETRIX    5773 7100139340       5760

ln142196:~ # sympd list
Symmetrix ID: 000190300571

Device Name          Directors                   Device
-------------------  ------------------  -------------------------------
------              -------              ------
Cap                Sym  SA :P DA :IT  Config      Attribute  Sts (MB)
-------------------  -------  ------  -------  -------------------  ------
------
/dev/dasdc 0138 02C:0 15B:D4 2-Way Mir N/Grp’d RW 6
/dev/dasdd 0139 02C:0 02A:D6 2-Way Mir N/Grp’d RW 6
/dev/dasde 013A 02C:0 15B:C5 2-Way Mir N/Grp’d RW 902
/dev/dasdf 013B 02C:0 15A:D5 2-Way Mir N/Grp’d RW 902
```

Configuring EMC Symmetrix Arrays
for Linux on System z
Applied Technology
There are no special requirements for SE for FCP devices on SuSE. Here is how SCSI FBA devices appear on the Linux virtual machine.

```bash
ln142197:/sys/bus/ccw/devices # syminq
```

<table>
<thead>
<tr>
<th>Device</th>
<th>Type</th>
<th>Vendor</th>
<th>ID</th>
<th>Rev</th>
<th>Ser Num</th>
<th>Cap (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda</td>
<td>EMC</td>
<td>SYMMETRIX</td>
<td>5773 71002C4000</td>
<td>17677440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/sdb</td>
<td>EMC</td>
<td>SYMMETRIX</td>
<td>5773 71002C5000</td>
<td>17677440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/sdc</td>
<td>EMC</td>
<td>SYMMETRIX</td>
<td>5773 71002C6000</td>
<td>17677440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/sdd</td>
<td>EMC</td>
<td>SYMMETRIX</td>
<td>5773 71002C7000</td>
<td>17677440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/sde</td>
<td>EMC</td>
<td>SYMMETRIX</td>
<td>5773 71000A0000</td>
<td>919680</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion**

The costs associated with IT resources are increasing every day — space, power, cooling, and people as well as the cost of the capital assets can be considerable. Balancing the costs with the ever-increasing demand for more processing power and storage capacities can be a daunting task for any company. Not to mention the complexities of managing the hundreds if not thousands of systems running the business-critical applications with storage capacities that scale into the petabytes.

These challenges have led organizations to explore virtual technologies as a way to control costs and reduce management complexities. Linux on System z provides data centers with an alternative that provides better input into the overall decision process of an organization’s IT resources. It allows virtual environments to take the place of physical servers on a platform that already runs the mission-critical applications in the enterprise. This union of open systems operating systems on z/Series hardware provides a consolidation option that is flexible and highly reliable. It reduces the costs with a smaller footprint on a platform that is already running in the data center. The virtualization that z/VM provides with its Linux guest virtual machines offers rapid deployment with base characteristics and easy expansion based on real needs so adding servers has no additional hardware or facilities costs.

Combined Linux on System z with the Symmetrix storage system allows for easy integration into the Linux environment. Whether the drive choice is ECKD or FBA, or the connectivity is FICON or FCP, the Symmetrix flexibility with support for both the mainframe and open systems coexisting in the same array provides rapid deployment options and simplified administration at the storage level. It presents a single set of storage administration tools that reduce the complexities inherent in a mixed platform environment.

The Symmetrix array has a proven track record of scalability, stability, and high performance that offers the best fit into the Linux on System z environment whether it’s running as a guest virtual machine under z/VM or standalone in a System z Logical Partition (LPAR).

By leveraging the document considerations from this paper, organizations can quickly integrate an Enterprise Linux (Red Hat or SuSE) solution on the world’s most reliable and highest-performing virtual platform and storage system — z/VM on the IBM System z with EMC Symmetrix. The flexibility that the Symmetrix array, Linux on System z, and z/VM provides makes for a viable innovative solution now and well into the future.

**References**

EMC-related documents include:
- EMC Solutions Enabler Version 7.0 Installation Guide
- Linux on IBM System z: RHEL 5.x and SLES 10.x Installation and Configuration Guide
- Native Multipath Failover Based on DM-MPIO for v2.6.x Linux Kernel and EMC Storage Arrays Configuration Guide
Third-party related documents include:

- Linux on zSeries and S/390: Distributions (IBM form number SG24-626)
- Introduction to the New Mainframe z/VM Basics

For a complete list of related reference material, refer to the following websites:

- [http://www.novell.com/linux/mainframe](http://www.novell.com/linux/mainframe)
- [http://linuxvm.org/](http://linuxvm.org/)