Solutions Enabler & its SYMAPI Force:
A Pioneer in a Remote Software Revolution

EMC Proven™ Professional Knowledge Sharing 2008

Ramiro A. Canovas
Regional Software Support Specialist
EMC Corporation
Canovas_Ramiro@emc.com

Table of Contents

Comprehend

Monitor & Manage

Create

Secure

Simplify

Optimize

Summary

Disclaimer: The views, processes or methodologies published in this compilation are those of the authors. They do not necessarily reflect EMC Corporation's views, processes, or methodologies.

Introduction

Ten years ago, who would have thought that hardware availability, efficiency and management would become dispensable without the aid of software? Virtualization, wireless advancements and remote management software have expanded Information Technology's frontier. Solutions Enabler (SE) was a pioneer that participated to this expansion. Through its Symmetrix API's (SYMAPI) strength, it has emerged and proven its value by monitoring, controlling and managing a customer's complex storage.

Solutions Enabler not only wears the Symmetrix® remote software management "attire" but it also serves as a Gateway to Symmetrix error reporting and to a Graphical User Interface (GUI) software that most customers are familiar with: EMC ControlCenter®. Solutions Enabler allows customers and technical field personnel to apply the following solutions:

✍ Discovery of Symmetrix and CLARiiON arrays (yes! CLARiiON)

✍ Monitoring and management of Symmetrix arrays

✍ Masking solutions for Symmetrix arrays

✍ Symmetrix configuration changes

✍ Security and access restrictions to a Symmetrix array

✍ Monitoring and management of Business Continuance Volumes (BCV) and Remote Data Facility (RDF) devices

This article will not be another product guide nor "inflate" a product to obtain a sale goal. It is a combination of field experience, customer examples and demonstrations describing how useful Solutions Enabler can become. Moreover, it will provide best practices that allow you to "squeeze the juice" out of such a resource-full remote software product. I will make frequent comparisons between Solutions Enabler and EMC ControlCenter. The preferred method will be depicted by the following icon: ☿ (based on field experiences and troubleshooting facts)**.**

I will also give extra troubleshooting tips and these will be depicted by the following icon:✎

Sit comfortably, relax and submerge into the SYMAPI world!

Concepts of Solutions Enabler and its SYMAPI

Before getting into details that pertain to Solutions Enabler, it's important to understand the simple concept of the software and its interaction with the customer's IT environment. **Figure 1** provides an introductory step:

After Solutions Enabler

Before Solutions Enabler

After Solutions Enabler

✓ Create Devices.
✓ Mask Devices.
✓ BCV Operations.
✓ RDF Operations.
…

Before Solutions Enabler

✓ Create Devices.
✓ Mask Devices.
✓ BCV Operations.
✓ RDF Operations.

SYMAPI
DB

Fibre Channel

I/O SCSI Commands

Fibre Channel
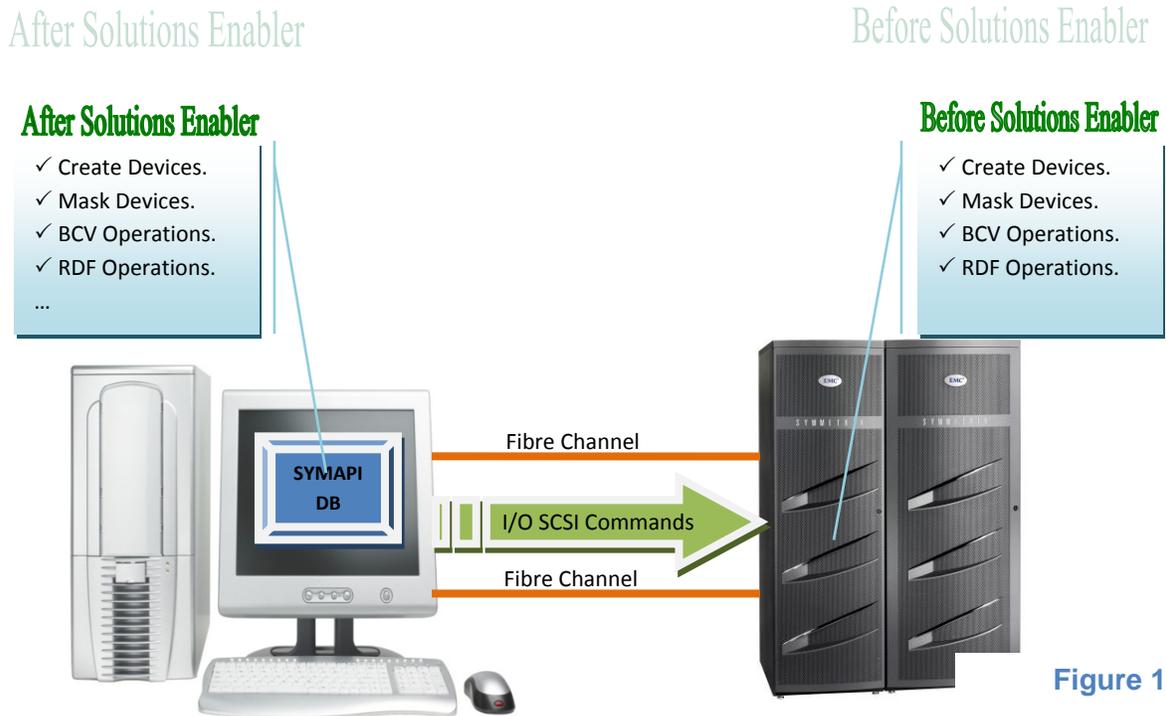
Figure 1

What is Solutions Enabler?

SE is a host-residing software that communicates with EMC Symmetrix and CLARiiON arrays through an Application Program Interface (API) called Symmetrix API.

What is SYMAPI and how is it useful to SE?

SYMAPI is a library of commands that uses system calls (syscalls) that generate low-level I/O SCSI commands to a customer's array. Be careful not to get confused by the "low-level I/O SCSI" fragment as the communication that "gives life" to these commands is physically possible only through at least one Fibre Channel connection to the array.

What exactly are syscalls?

Syscalls consist of two SCSI commands: Command 3B (Write Buffer) and 3C (Read Buffer). These commands are not designed to write/read data to the media but rather to a buffer in the controller memory. The SCSI standard says they are for diagnostics and vendor specific purposes. EMC uses them to transfer commands and their results for its software products, in this case SE.

Where does the user run the SE commands from?

All SE commands are run through the Command Line Interface. This makes SE universal as it can be installed and used in any host type:

- Windows (DOS Prompt)
- SUN/Solaris
- IBM AIX
- Linux/SUSE
- Z/OS Hosts
- TRU64 Hosts
- OpenVMS
- HP-UX

If it can be installed in any host type, what prevents excessive SE inquiries to the arrays?

SE creates a database that resides in the host; it possesses all the relevant information regarding the array. This database is called the SYMAPI Database or Symmetrix Configuration Database and avoids excessive inquires to the array from multiple hosts.

Birth, development and apogee of Solutions Enabler

In 1997, the sole purpose was to create a command line interface to monitor and manage the Symmetrix array. At the time, there were only about 12 commands that would allow features such as Base, TimeFinder®, SRDF®, and Symmetrix Resource Manager (SRM). Even the easiest change (except front-end device mapping) required EMC field personnel.

After a name transition in 2001 to WideSky SE, Solutions Enabler gained its strength and value when configuration manager was added as a feature to reach almost 46 useful commands in the latest version 6.5. Now, SRDF and TimeFinder features are among the strongest points.

If you have been working with Symmetrix arrays for more than ten years, you remember having to change a bit flag in a Symmetrix array. Only an EMC field service provider who had submitted a Change Control Approval (CCA) could do it. This CCA would be approved in approximately a week; then the EMC representative would have to be onsite and request an outage time window from the customer. In summary, such a simple change required a lot of time and several steps.

I conducted a survey in the southeast division asking several Customer Service field engineers what they thought about the old way of managing a Symmetrix array and the advantages of SE as remote software:

Mitch Gass - EMC Customer Engineer for 9 ½ years – Huntsville, Alabama

"I remember having to spend hours setting up SRDF groups and creating them in the separate bins. Nowadays, I use SE for Lun changes and especially for SRDF configurations. In my opinion SRDF is the biggest advantage of using SE. Takes a lot less time and configures both boxes at the same time."

Mike Rockwood – EMC Customer Engineer for 9 years – Atlanta, Georgia

"The biggest advantage I saw with SE was the reduced time it takes to implement simple changes for the customer. No week long wait while the change control process is underway. Greater flexibility for the customer in scheduling changes as well. Lastly but not least, reduced workload for Customer Service."

Glenn Clark – EMC Customer Engineer for 7 years - Raleigh, North Carolina

"The most obvious advantage of the using SE/ECC is the decrease in CCA activities; more specifically in assignment changes. We, meaning CS, used to do all the assignments changes for all customers that were not on SANs or did not use SE/ECC. We practically lived in datacenters and the OT was off the chart"

How to obtain the necessary Symmetrix information through Solutions Enabler

Necessary information is information we need to monitor and manage a Symmetrix.

I always like to ask what characteristics effective software should have. One of them is providing the exact same information that you would obtain if standing right in front of the storage system; SE delivers. With Solutions Enabler, it's always important to be as specific as possible when inquiring for information. The best way is to add as many flags as you can in your SE commands.

Where creativity meets information

Once you fully understand how to efficiently use SE commands, the rest is up to the imagination as each command has numerous flags and multiple combinations. Every time I am with a customer, I constantly repeat "it's all about understanding the format." I will explain what the format is and will use an example to help you understand.

In SE, we always have 4 main components in a full command:

1) Main command
2) Action
3) Flag
4) Content of flag

Let's take a simple symdev command as an example (its usage is explained below)

$$\text{Symdev}_{1} \quad \text{list}_{2} \quad \text{--sid}_{3} \quad \text{0053}_{4}$$
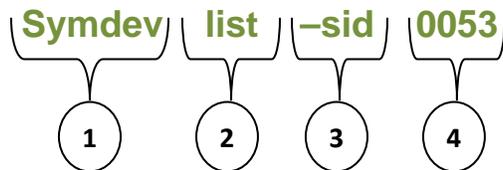
Sequence is usually not important as long as the main command is the first one typed.

The above symdev command is one of the most useful commands when it comes to displaying Symmetrix device information.

Now, what happens if we are trying to display devices in a specific Front end Adapter (FA) or port? The above command will display redundant information and might make the output confusing. With this said, the best way to use the above command would be:

- symdev list –sid <Symmetrix ID> -SA <Symmetrix FA> -p <Symmetrix port>

                   Flag   Flag   Flag

For example:

 # symdev list -sid 0968 -sa 14a -p 0

Symmetrix ID: 000190100968

Device Name		Directors	Device				
			Cap				
Sym	Physical	SA :P DA :IT	Config	Attribute	Sts	(MB)	
0040	/dev/rhdisk4	14A:0 02C:D12	2-Way Mir	N/Grp'd	VCM RW	90	
0049	Not Visible	14A:0 02D:D11	2-Way Mir	N/Grp'd	RW	3	
004A	Not Visible	14A:0 01A:D11	2-Way Mir	N/Grp'd	RW	3	
004B	Not Visible	14A:0 16C:C11	2-Way Mir	N/Grp'd	RW	3	
004C	Not Visible	14A:0 15B:C11	2-Way Mir	N/Grp'd	RW	3	
004D	Not Visible	14A:0 02B:D13	2-Way Mir	N/Grp'd	RW	3	
004E	Not Visible	14A:0 01C:D13	2-Way Mir	N/Grp'd	RW	3	
004F	Not Visible	14A:0 02D:D13	2-Way Mir	N/Grp'd	RW	3	
0050	Not Visible	14A:0 01A:D13	2-Way Mir	N/Grp'd	RW	3	
0081	Not Visible	14A:0 01A:C12	2-Way Mir	N/Grp'd	RW	2157	
0082	Not Visible	14A:0 02D:C12	2-Way Mir	N/Grp'd	RW	2157	
0083	Not Visible	14A:0 15B:D4	2-Way Mir	N/Grp'd	RW	2157	
0084	Not Visible	14A:0 16C:D4	2-Way Mir	N/Grp'd	RW	2157	
0085	Not Visible	14A:0 15D:D4	2-Way Mir	N/Grp'd	RW	2157	

SE allows you to go in depth when inquiring the Symmetrix. In these sort of inquiries, the –v flag (standing for verbose) plays a big role. Some of the strongest examples are:

- symdev show <dev> - displays Symmetrix device characteristics.
- symcfg list –v   - displays Symmetrix array characteristics.

When dealing with masking issues such as servers not being able to see a specific device, the symdev show command can be very useful as it will point out the state of the device throughout all FAs that the device is mapped to.

How can you monitor resources such as GKs or make sure that the SYMAPI DB is not being used? The command below is not very popular, but very useful:

- symcfg list –semaphores

#symcfg list -semaphores

```
          S Y M A P I   S E M A P H O R E S
               Lock Proc.
  ID    Key       State   Type Wait      Lock          Full Path Specification
 ------ ---------- -------- ---- ----- ------------------------------------------------------------
  13 0x450011d8 Unlocked GK          0                /dev/rhdiskpower0
 11 0x45001068 Unlocked GK          0                /dev/rhdiskpower1
 14 0x450011db Unlocked GK          0                /dev/rhdiskpower2
 12 0x450011d9 Unlocked GK          0                /dev/rhdiskpower3
 15 0x4500106c Unlocked GK          0                /dev/rhdisk1
 9 0x45001077 Unlocked GK          0                /dev/rhdisk2
 16 0x450011bc Unlocked GK          0                /dev/rhdisk9
 131079 0x45002848 Unlocked DB      0                /var/symapi/db/symapi_db.bin
 131080 0x45002847 Unlocked FILE    0                /var/symapi/config/symapislck
```

✔ If you ever obtain the following output after issuing a SE command: "The gatekeeper device (while using the Base Daemon) has an error", first verify that you have the necessary GKs and then issue the above command again to make sure they are not being used (locked).

As we can see, there is an extensive list of commands that will give us Symmetrix information but with a little bit of creativity, knowledge of the format and the type of activity to be performed there is really nothing that should stop you from efficiently using the SE commands.

How to efficiently prepare a Symmetrix environment for remote control

Establishing a consistent and secure connection to the array is most important step to efficiently manage an IT environment remotely through software.

It is important to understand that all it takes to efficiently manage a specific Symmetrix is just one host that can access its databases. This situation is explained in detail in the Secure section (Limiting access to a Symmetrix VCM).

We will first "tackle" the physical requirements: fibre connectivity! Any host that will be either issuing SE commands or attempting to see FC devices will need to be connected through Fiber to the array.

Secondly, establish communication between servers and array. It's vital to establish communication between the servers' HBAs and the Symmetrix FA's through zoning. This article will not go into details on zoning practices or steps but it's important to mention.

We will now get to what concerns us most, how to establish communication to the array through SE. The VCM Database device and the Gate Keepers (GK) should be seen first. Without VCM, there would not be access to the Symmetrix Information, and without GKs no resources to issue the SE commands.

Here is one command we should always remember:

- symcfg discover

This command always searches and refreshes all the Symmetrix information by bringing all the relevant information into the SYMAPI DB.

🔧 If you ever issue a symcfg discover and the output is: "no devices were found", the server is not seeing the VCM device, pointing to a physical issue (cables), zoning issue or server configuration.

In a new array, the VCM will be uninitialized so we will issue our first command:

- symmaskdb init –sid <Symmetrix Id> init –file <filename>

🔧 In the above command, the –file flag allows a second layer of protection in case the VCM has already been initialized and had masking entries. It will create a file that can be used as a backup to restore from. When you initialize the VCM DB, you are basically wiping everything inside it, so use it with caution.

The symcfg discover, when working correctly, will not display any output. The best way to determine if all the information was gathered correctly would be to proceed to our next step: verify that all your server's HBAs are communicating and bound correctly to a Symmetrix FA.

You can do so by issuing:

- symmask discover hba
- symmask list hba

Finally, verify that we can start communicating with the correct Symmetrix

- symcfg list

This output will help you to understand the commands mentioned above:

 # symcfg discover

This operation may take up to a few minutes. Please be patient...

 # symmaskdb init -sid 0968 -f VcmBackup

The database on Symmetrix 000190100968 is a valid database and a backup file was created. Completing this action will wipe out all records in the database.

WARNING: All hosts will lose access to their devices on Symmetrix 000190100968.
Initialize Symmetrix SymMask database on Symmetrix 000190100968 (y/[n])? y

 # symmask discover hba

Symmetrix ID   : 000190100968
Device Masking Status: Success

Identifier   Type   User-generated Name
---------------- ----- --------------------------------
10000000c932913c Fibre safeaix2/10000000c932913c

 # symmask list hba

Identifier   Type   Adapter   Physical Device Path   Dir:P
---------------- ----- --------------- ----------------------- -----
10000000c932913c Fibre 20-60   /dev/rhdisk4   14A:0

symcfg list
               S Y M M E T R I X

SymmID	Attachment	Model	Mcode Version	Cache Size (MB)	Num Phys Devices	Num Symm Devices
000190100968	Local	DMX3-24	5771	98304	1	1027
000187400727	Remote	DMX1000S	5671	16384	0	963

Masking Symmetrix Devices to a specific Host

When we talk about masking, we are referring to the presentation of FC devices to a specific host's Host Bus Adapter (HBA). Before SE's development, masking was only done through BIN File changes and it did not allow much flexibility. With SE, masking has become more secure.

When we deal with masking, we always have to consider the VCM DB. Without it, there would not be any masking entries and no assignments to hosts. We always recommend that you double check the VCM DB before, and especially after each masking activity.

<u>Before</u>: Is the VCM DB initialized? If it is, is it empty or does it already have masking assignments?

<u>After</u>: Was the masking assignment correctly done? Did I pick the right devices?

Answer these questions using this command:

- symmaskdb list –sid <Symmetrix ID> database

 It's important to understand that the VCM DB device can be mapped to different FAs. With this said, one of the most frequent user mistakes is choosing to mask the right device to the right HBA through the wrong FA. The above command can clear any doubt and point to the right troubleshooting direction.

How can we efficiently mask a specific device to a host?

1) Select the devices to be masked using the information described in "Obtaining the necessary information through SE." Be careful not to pick devices that do not have an address in the FA that your servers are attached to (not mapped). Otherwise, all the below steps will be in vain.
2) Avoid using WWPNs when dealing with masking. Why? In a big and complex customer environment, WWPNs (thus numbers) can get very confusing. I recommend assigning an alias to each one of your WWPNs.
   o symmask –sid <Symmetrix ID> -wwn <wwn of HBA> rename <name>
3) Once you properly select your devices and the server's HBAs that will be assigned, it's time to do simple masking!

When doing masking, SE will prompt you with a warning if the specified device is already masked to another HBA. <u>However</u>, if you mask a device to a wrong FA, it will also warn you but there will not be a prompt – meaning that it will still proceed with the action – so be aware!

4) Finally, refresh every connected FA with up to date information from the VCM DB. This command refreshes the host-related profile data in the Symmetrix array only (compared to the host DB that is refreshed through the symcfg discover command):
   o Symmask refresh

EMC ControlCenter versus SE: With EMC ControlCenter, masking can be user friendly through its GUI and you can perform all the above steps in one useful wizard (including refreshing the VCM DB). Please remember that it takes longer to assign the devices than it would in SE. ♉

How can SE assist in troubleshooting?

SE can be a very powerful troubleshooting tool when it comes to issues with missing or disappearing devices. As a Regional Software Specialist, it's one of the first tools I reach for when starting to analyze an issue. EMC Reports (for Windows) and EMC Grabs (for Unix) are two potent scripts that use SE to acquire logs from the server with issues; in fact all the script does is run SE commands in the server and save the output in a txt file. Gaining familiarity with SE commands is all it takes to know where to go to start troubleshooting.

For example, customer X states that she masked Symmetrix devices to an IBM AIX server and she sees only 1 of 2 devices. These devices are devices 4a and 4b (which is the one missing). Customer's server is zoned to FA 14a port B and FA 7a port B. Customer runs an EMC grab in the host and sends it for analysis. Without looking at the EMC report, what would be the first log you would be looking for? If you said the one related to the VCM DB you are in the right track. Remember that logs are merely SE commands so you would be looking for symmask_list_database.txt log. By looking at the log:

```
Symmetrix ID: 000000006196
Last updated at: 04:58:00 PM on Tue Mar 25, 2007
Director Identification: FA-14A
Director Port: 1
User-generated
Identifier      Type   Node Name                    Port Name          Devices
--------------- ----- ------------------------------- ---------
10000000c924e04a Fibre  CustomerX_AIX       14a0         4a

Director Identification: FA-2a
Director Port: 1
User-generated
Identifier      Type   Node Name                    Port Name          Devices
--------------- ----- ------------------------------- ---------
10000000c9238053 Fibre  CustomerX_AIX       14a0         4b
```

We can see that the masking has been done incorrectly. The customer masked the right device (4B) to the wrong FA (2aB is incorrect) when both 4a and 4b should have been masked to FA 14A port B (1)

This is just a simple example but, with SE, you can troubleshoot as deep as FA configuration, device mirroring, device mapping, and many other details on the Symmetrix and its components.

Let us look at one more complex example. Customer Y started having issues with one of his RDF devices but he cannot recall when the problems started. Once again, SE commands can lead us to the solution. First, we will need to find out if and when there were any first RDF events going on in the Symmetrix. By issuing a *symevent list –sid <Symmetrix ID> -v* we will see all the events that took place in the Symmetrix Array (remember to be as specific as you can by using the flags).

```
# symevent list –sid 0968 –v –start 09:00 –end 17:00

Event at Sun Jan 13 00:54:46 2008:
    Reporting Director: FA-14A
    Source         : Symmetrix
    Category        : RDF
    Severity        : Informational
    Numeric Code     : 0x0018
    Event Code Symbol: RDF2_DEVICE_NR
    Description       : One of the RDF2 devices was found to be Not Ready

Event at Sun Jan 13 00:54:56 2008:
    Reporting Director: FA-7A
    Source         : Symmetrix
    Category        : RDF
    Severity        : Informational
    Numeric Code     : 0x0018
    Event Code Symbol: RDF2_DEVICE_NR
    Description       : One of the RDF2 devices was found to be Not Ready
```

The above output did not only point us to a timeframe but also to where the root cause might be: An RDF2 device in FA 14A (which is being used by the customer) and 7A (which is not used so we will leave that to the side). Now we can perform a *symdev list* to verify which R2 is in a 'not ready' state. Once we find the device, we can use a SE RDF command such as *symrdf list* to list all RDFs and its relationships. After finding the SRDF relationship and narrowing down the device, we can bring it ready again with a *symdev ready <Symmetrix device>* command.

EMC ControlCenter versus SE: EMC ControlCenter allows you to make a device user not ready, user write disable, user ready or user read write enabled with a simple right click.   SE will need the correct syntax and format to execute the command. My choice: EMC ControlCenter ♉

EMC ControlCenter versus SE: With EMC ControlCenter, troubleshooting through the console GUI interface might be time consuming and tedious. Even though the VCM information is there, there are several screens you will have to browse through to finally have everything you need in front of you. With SE, all it takes is a simple command. My choice: Solutions Enabler ♉

These are just a few examples that portray how SE can be used as a remote troubleshooting software tool.

Symmetrix Configuration Changes

We are now entering the "apogee" phase of SE. As stated in one of the earlier sections of the article, configuration changes were always done through the BIN file in front of the array and after lengthy CCA approvals. What shortened those CCA approval times was and still is the fact that SE has very strict rules and checks that will prevent the customer or the field service personnel from making mistakes when performing a configuration change (config change as called in the field). I will explain the rules once we get more familiar with what sort of configuration changes can be performed with SE.

Configuration changes consist of a "home made" script and one main SE command. Every configuration change is performed against that script. The SE command is *symconfigure*.

There are 3 main stages in a configuration change that can be applied to every SE config situation.

1) Preview: In this step, the software checks for any type of syntax errors and the accuracy of each change.
2) Prepare: In this step, the software checks the actual change and makes sure that the Symmetrix is in the correct state to perform the configuration activity. Moreover, it will ensure that it stays in the same state after the change.
3) Commit: This is the point of no return as it executes the change.

To illustrate the script, the command and the rules:

a) User creates a script called create_meta.cmd. This script will contain the configuration activity (in this case create a Meta) that will follow the correct format (otherwise it will fail in the Preview stage!).
b) User will perform a *symconfigure –sid 0968 –f create_meta.cmd commit*

🔧 If you decide to type "commit" as the action, it will <u>still</u> go through the preview and prepare steps prior to doing a commit. What is the point of having all 3 steps as separate actions? In case you decide to check for the script syntax and Symmetrix readiness without making the actual commit!

c) Script will go through a Preview, then a Prepare and finally the point of no return: a Commit.

```
# symconfigure -sid 0968 -f create_meta.cmd commit

Execute a symconfigure operation for symmetrix '000190100968' (y/[n]) ? y

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    Processing symmetrix 000190100968
    Performing Access checks.................................Allowed.
    Checking Device Reservations.............................Allowed.
    Submitting configuration changes.........................Submitted
    Locking devices..........................................Locked.
    Validating configuration changes.........................Validated.
    Initiating PREPARE of configuration changes..............Queued.
    PREPARE requesting required resources....................Obtained.
    Step 013 of 017 steps....................................Executing.
    Step 015 of 017 steps....................................Executing.
    Local:  PREPARE..........................................Done.
    Initiating COMMIT of configuration changes...............Queued.
    COMMIT requesting required resources.....................Obtained.
    Step 019 of 118 steps....................................Executing.
    Step 024 of 118 steps....................................Executing.
    Step 024 of 118 steps....................................Executing.
    Step 024 of 118 steps....................................Executing.
    Step 024 of 118 steps....................................Executing.
    Step 069 of 152 steps....................................Executing.
    Step 069 of 152 steps....................................Executing.
    Step 101 of 152 steps....................................Executing.
    Step 103 of 152 steps....................................Executing.
    Step 104 of 152 steps....................................Executing.
    Local:  COMMIT...........................................Done.
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

EMC ControlCenter versus SE: With EMC ControlCenter, configuring can be very user friendly because, like masking, it has a configuration wizard. It is a great tool for novices as it shows what can be done; Solutions Enabler leaves it up to the user to know or investigate the software limitations. On a more advanced level, SE provides a deeper level of detail when it comes to the 3 steps. If a configuration change fails in EMC ControlCenter, the error will not be too descriptive; SE that provides the user with more details. Last but not least, let's not forget that EMC ControlCenter uses SE so it will be an extra hop for every command. My choice: Solutions Enabler ☿

The SE configuration feature has many sub-features that make it an influential remote tool. It is possible to monitor and even abort (use with care!) any configuration change that is going on in the Symmetrix. This can be done regardless of the host from which you are running SE, as long as you are seeing/sharing the same VCM DB.

Configuration change rules

Giving more control to the customer

The SE configuration feature allows the customer to fill a role that, until recently, required a hardware provider. Almost anything related to configuration changes can be done through this useful software. I say <u>almost</u> everything because activities such as code upgrades can only be done through the service processor in the Symmetrix.

What can be done?

⌨ Create and delete all type of devices: BCV, RDF, 2-WAY, VDEV, etc.

⌨ Convert any type of devices

⌨ Change device characteristics

⌨ Map and unmap devices (Give a device an address and take it away in the FAs)

⌨ Perform any type of RDF activity to the devices

⌨ Create, configure and convert Meta devices

⌨ Set port characteristics

⌨ Set Symmetrix overall attributes

Limiting the user for a safer change

SE rules allow a smooth and safe configuration change. These rules allow an automatic CCA without needing someone to make sure that the rules are followed.

With the advancement of the Symmetrix Microcodes, the rules have been changing, thus the below rules will differ depending on the Symmetrix type and code.

<u>These rules apply to enginuity 67 and 68 in Symmetrix 3000, 4000, and 8000 families:</u>

[1]"® **Same disk**: Two mirrors of the same device should never reside on the same disk (for example, 1a:C0)

® **X/Y bus:** Two mirrors of the same RAID 1 device should never reside on the same memory bus (for example, 1a with 3a).

---

[1] EMC Solutions Enabler Symmetrix Array Controls CLI guide. EMC Corporation 2007.

® **Dual-Initiator**: Two mirrors of the same RAID 1 device should never reside on the same DA and dual-initiator pair (for example, 1a with 2a)."

These rules apply to enginuity 69/70 and above in Symmetrix DMX family:

[2]"These rules are relative in manner. Optimizer can improve or maintain protection, but never decrease level of protection. However, some restrictions should always be followed.

® **Same disk:** Two mirrors of the same device should never reside on the same disk (for example, 1a:C0).

® **Same port:** Two mirrors of the same device should never reside on the same SCSI port (for example, 1a:C).

® **Same loop:** All mirrors of the same RAID 1 device should never reside on the same disk loop (for example, 1a:C with 16a:C).

® **Power zones:** All mirrors of the same RAID 1 device should never reside on the same power zone.

® **Same DA:** All mirrors of the same RAID 1 device should not reside on the same DA (CPU) (for example, 1a).

® **Same slot:** All mirrors of the same RAID 1 device should not reside on the same director (slot) (for example, 1a with 1b).

® **Dual-Initiator:** All mirrors of the same RAID 1 device should not reside on the same dual-initiator pair (for example, 1a with 16a)."

EMC ControlCenter versus SE: When it comes to rules, it's always good to know them but it's time consuming to figure them out by trial and error. EMC ControlCenter will show what can be done, so anything that is not specified or displayed is not permitted by the rules; period. With SE, you can try to perform a configuration change several times until you realize that you are violating a rule (although the error might specify the reason). My choice: EMC ControlCenter ☿

---

[2] EMC Solutions Enabler Symmetrix Array Controls CLI guide. EMC Corporation 2007.

Limiting access to a Symmetrix VCM DB

Redundancy is good but it's always advisable to avoid excess.   What would happen if we have 15 servers fiber attached to a Symmetrix array and they all have access to make changes to the VCM DB? I will list just a few of the problems:

💣 Excessive sys calls to the Symmetrix database

💣 Unauthorized access to the VCM database

💣 Fewer control on changes made to the Symmetrix and VCM Database

💣 Excessive masking mistakes

There are three solutions that I will propose to the above situations. But first, let's try to imagine the ideal scenario:

2 Fibre Connections

AIX

SUN

VCM
DB

W2K
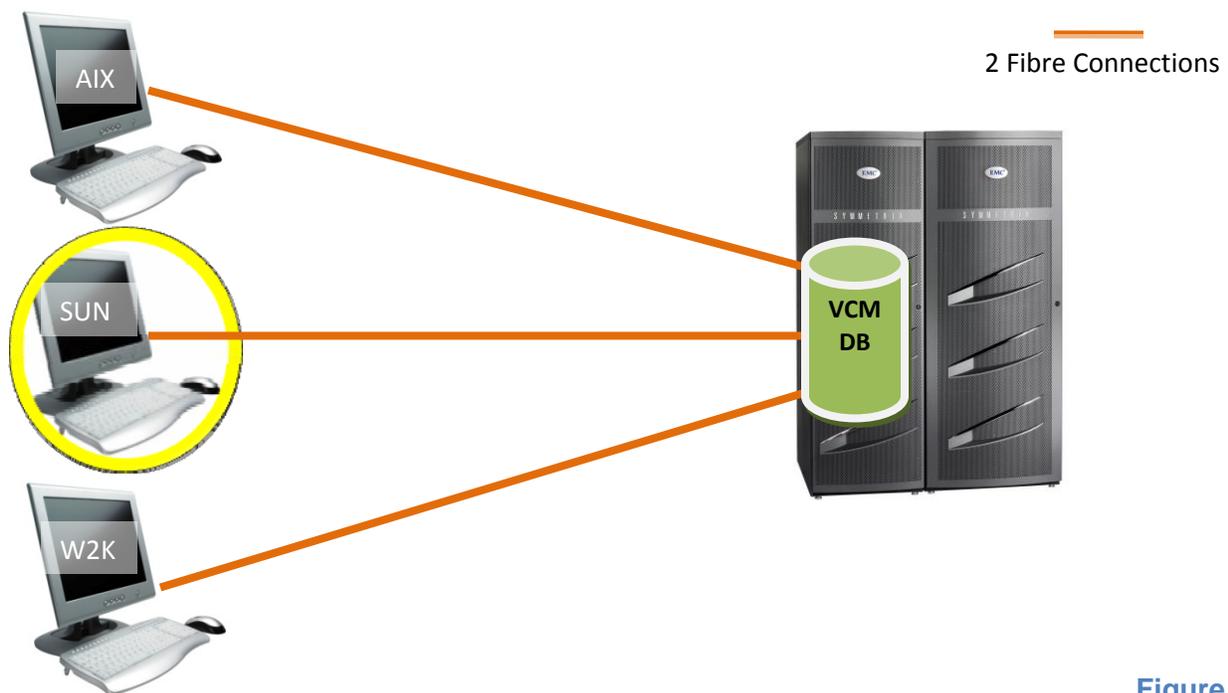
Figure 2

There are 3 different servers in Figure 2: 1 IBM AIX, 1 SUN and 1 Windows Server. They should all have fiber connectivity to the array if they are going to have FC devices presented to them. Because they all share the same VCM DB, the user will really need only one server (maybe two for redundancy) to have access to it, thus allowing only that specific server to make changes to the Symmetrix. In this example, the SUN system will be able to mask the devices for the rest of

the servers and will be responsible for making the configuration changes that might or might not affect the Windows and AIX.

What are the solutions?

✍ **VCMDB_restricted_access**: In older codes, you can change a Symmetrix overall configuration setting called VCMDB_restricted_access. This change is performed against the whole Symmetrix array and <u>not</u> against a specific port in a FA, and the possible values are either enabled or disabled. If disabled, then every host attached to a specific port in a specific FA will see and have access to the VCM DB just by zoning correctly (except the SUN host that requires extra steps). If enabled, only the servers that mask the VCM DB device prior to making the change will have access to it. In this way, you can have only one server with the VCM DB masked to it, providing a layer of security for the whole environment.
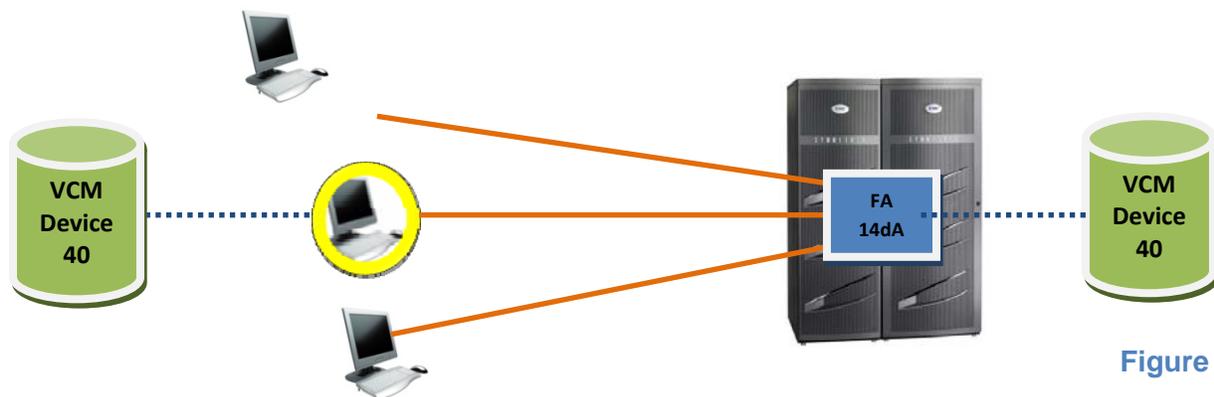

Figure 3

✌ **The power of assigning addresses:** The VCM DB is a device like any other in the Symmetrix. With that said, if you assign an address to the VCM to only one port in one specific FA, then any servers connected to other FAs will not have access to the DB. In this case, the solution is to attach only the SE Server Master (the server which will be making the changes) to that specific port/FA where the VCMDB has an address.

Figure 4

✍ **Access Control:** This is a feature of SE that will be explained in the next section. This concept is basically the creation of security groups that restrict access to the VCM DB.

Access Control Groups

SE software has different layers that allow different operations, the main layers are listed below (total of 13). Each requires a separate license:

Base: Allows running the main commands such as symcfg discover, symdev list, etc…

Device Masking: Allows masking operations

RDF: Allows RDF operations

TimeFinder: Allows Local Replication (BCV) operations

Config Manager: Allows performing configuration changes to an array

With so many layers, wouldn't it be smart to restrict the access depending on what is needed or being used? For example, a corporation has an enormous IT environment where there are three different IT departments who all need access to perform different SE operations:
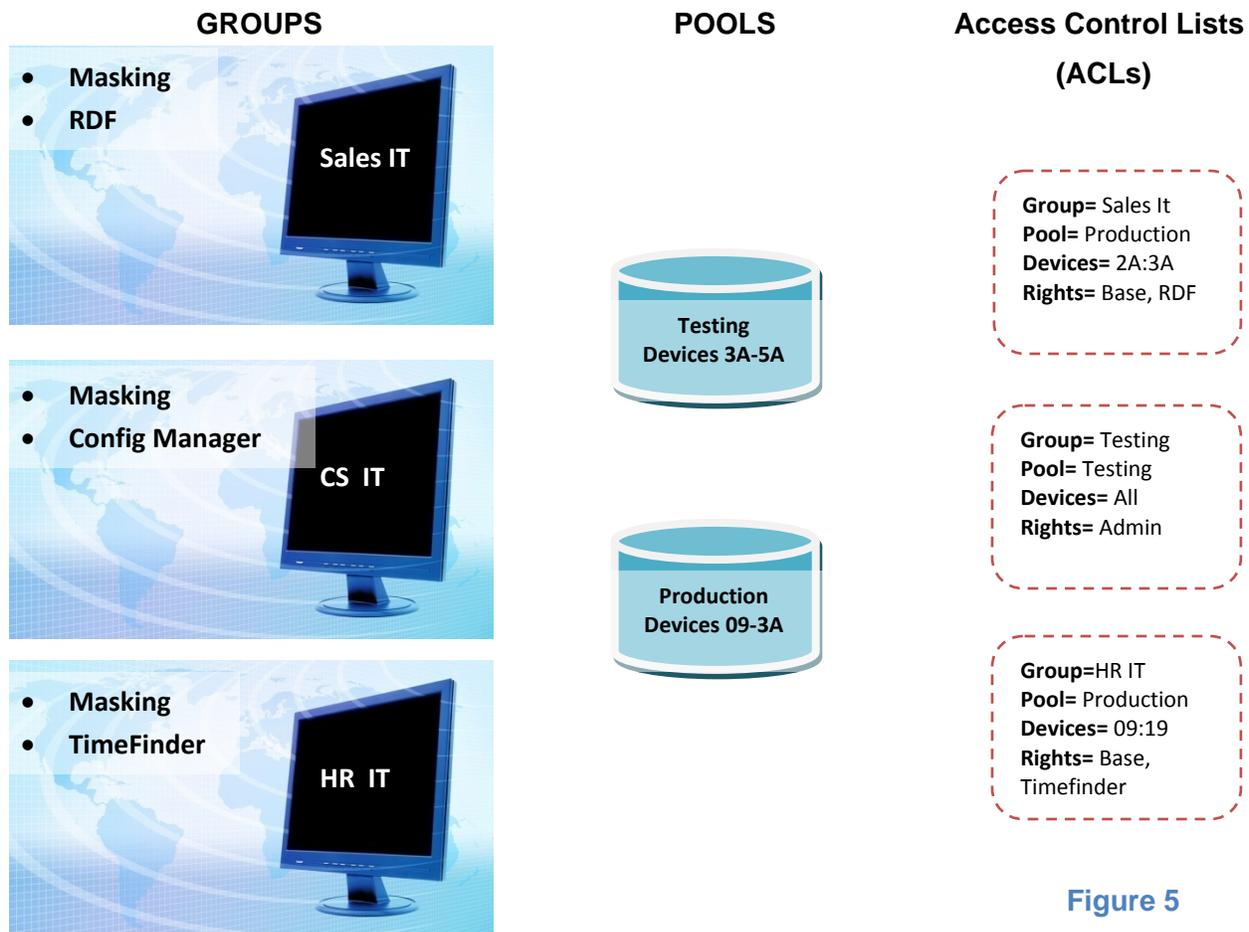

Figure 5

You can limit the Symmetrix usage with Access Control Lists (ACLs) according to each department's IT need. For example, the CS Department has full access to all the devices in the testing pool. Moreover, they have all rights to do any SE operation on the specified devices.

The SE enabler command that allows this is *symacl* which allows the administrator to "play god" in the environment. The steps to correctly create ACL are:

1) Create Access Control Groups (ACG)
2) Assign a Host ID to the specific host that will be part of the ACG
3) Create an Access Pool
4) Add devices into Access Pool
5) Grant rights to an Access Group

Security can be such that you can also apply Access IDs, also known as PINs, to several operations. For example, if someone is attempting to perform a configuration change, the user will not be able to do so unless they enter the PIN prior to the activity. ACLs in SE can be a very efficient tool to organize and control your IT environment.

EMC ControlCenter versus SE: In Parallel to ACG in SE, EMC ControlCenter provides a useful security layer. You can specify what type of role each user has and what limitations and capability users will possess in the environment. This tends to be very intuitive and valuable when having several people accessing the EMC ControlCenter console: My choice, EMC ControlCenter ♉

How does EMC ControlCenter utilize SE?

EMC ControlCenter is universal management software that is accessed through a GUI console. Even though EMC ControlCenter is able to manage several arrays such as CLARiiON, Celerra®, HP STORAGEWORKS, and other 3rd party arrays, the software's strongest benefit is monitoring and managing Symmetrix. How can this be done? Through SE! **Figure 6** will display SE power and influence in an EMC ControlCenter environment.
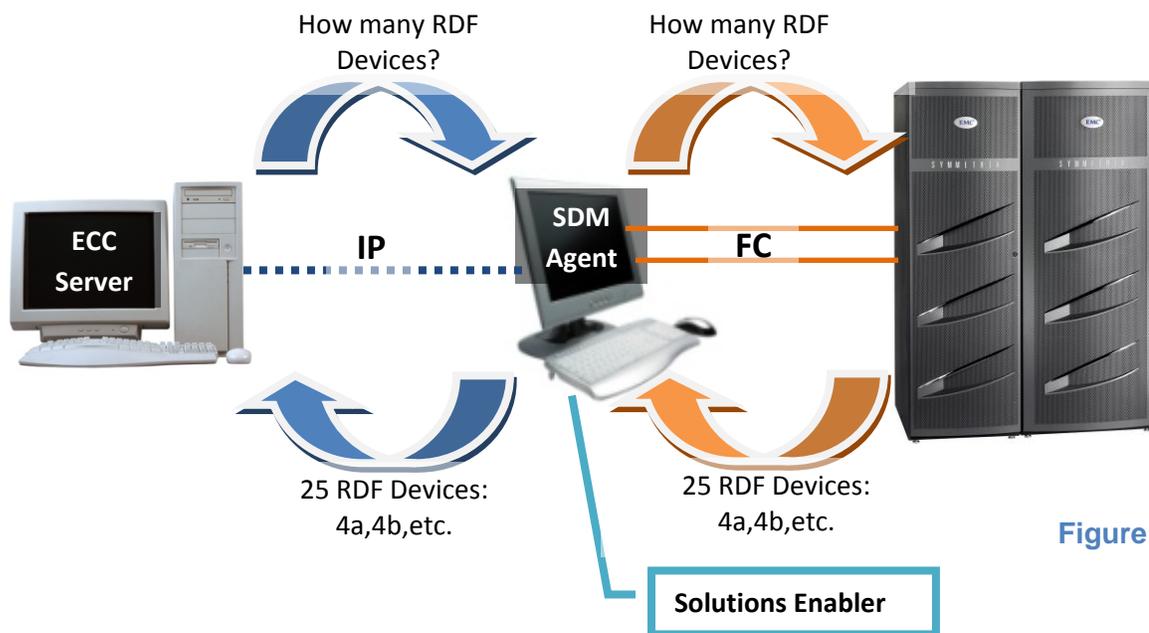

Figure 6

EMC ControlCenter software obtains all the required information through agents. These agents reside in different hosts that are IP connected to the EMC ControlCenter Server. SE takes all the applause in the case of Symmetrix information. Why? Every time EMC ControlCenter needs Symmetrix information, it uses its Symmetrix dedicated agents: Symmetrix agent and Symmetrix Device Masking (SDM) agent that communicate to the server through IP. These agents use the SYMAPI library to gather all the requested/required Symmetrix through a similar data process that a user would have when using SE as a standalone software.

Solutions Enabler: A family of three
Solutions Enabler has three different software packages that contain different features:

1) Full Solutions Enabler: If fully licensed, possess all control and monitoring operations
2) Monitoring Solutions Enabler: No control operations, merely monitoring
3) Solutions Enabler Repository Kit: Created to be installed in the EMC ControlCenter Server. The user can deploy SE as if it was an EMC ControlCenter agent to all managed hosts.

Best practices when using an environment with SE and EMC ControlCenter

Your best friends: The Gatekeepers (GK):
GKs are responsible for provisioning resources to the SE commands. Every time there is a low I/O SCSI command, the resources are expected to come from the GK. Because these devices have such importance in an SE environment, I recommend assigning 2 <u>dedicated</u> GKs per Symmetrix FA port.

For EMC ControlCenter, it all depends on the configuration. If the Storage Agent for Symmetrix is installed on the host, there should be at least 2 gatekeepers. Moreover, if the Symmetrix Configure commands are being used to manage the Symmetrix system, there will be a need for 2 extra GK (4 total). Other agents such as Common Mapping Agent and host agents do not require a gatekeeper as they do not deal with the Symmetrix through SE.

With EMC ControlCenter, it is even easier to find out if the environment is lacking GKs. If the host that is managing the Symmetrix (through the agents) does not have enough GKs, there will be an alert "Symmetrix <symm ID> has *count* gatekeeper(s) available" that will pop up meaning that the GKs assigned are not sufficient.

Redundancy, redundancy, redundancy…:

Redundancy plays a big role in an SE environment, especially when making configuration changes. In SE, redundancy paths make a configuration change online. For example, let's say that Customer Y wants to change the heterogeneous bit flag in FA 13b port a, the change will be online because customer Y should have, at least, dual paths going to FA 13b port a and FA 6b port a (that will stay online and will be able to access the devices).

In EMC ControlCenter, the redundancy is done through agents. Despite the fact that, in most scenarios, a single agent will be enough to gather managed object information. If the server that hosts the agents goes down, the agents will be inaccessible and unable to gather data. This is the main reason why you should always have at least 2 agents of each type in separate servers. In the case of a Symmetrix agent, the user has to make sure that both servers – with a Symmetrix agent each- are seeing the same Symmetrix (otherwise the redundancy will be worthless!)

If troubleshooting agents through EMC ControlCenter, the agents' view are the best windows that will allow you to determine where to start (make sure you choose show all). If an agent is down, it will point out to the server that hosts this agent. Most importantly, it will show you the version of the agent in comparison to the rest (allowing you to determine if it's a version incompatibility issue). By doing this, you are narrowing down the possible root causes of an agent malfunction.

Back them up and you will backup your job

Always backup your VCM before and after making any changes. It is vital that you back it up by issuing a backup command and not an initialize command (as it will also back it up prior to initialization)

symmaskdb backup –f <filename>   **symmaskdb init –f <filename>**

When you deal with Device Groups, BCVs and RDF, always back them up in case you need to mechanically replace an array or re-image a server. Remember that device groups are host specific, not Symmetrix.

Before and especially after any configuration change, make sure to run a *symcfg discover* to refresh all the Symmetrix information. In this way, the user will avoid relying on old configurations and avoid performing an incorrect operation.

EMC ControlCenter versus SE: Finding the correct command to backup the VCM or initialize it may be time consuming. With EMC ControlCenter, all it takes is to be in the right place and ask the right question: Where does the VCM device reside? The answer should be Symmetrix, and then all that is left is to right click and find the backup or initialize VCM option! My choice: EMC ControlCenter ♉

Summary

This article describes how a remote software tool such as Solutions Enabler changed the face of lengthy Symmetrix activities. Furthermore, it provides insight into a resourceful tool through examples, comparison with EMC ControlCenter and field experiences. Such robust and customizable software not only serves as a secure tool to monitor and manage the Symmetrix array, but also serves as a useful troubleshooting tool.

From simple activities such as displaying Symmetrix information to more complicated activities such as creating devices or changing FA bit flags, SE can be a remote solution to multiple IT obstacles that customers and EMC field personnel face every day.

Bibliography

📖 EMC Primus Database: solutions emc39096 & emc94039
📖 EMC Solutions Enabler Access Array Control guide – EMC Corporation 2007
📖 EMC Solutions Enabler Symmetrix Array Controls CLI guide - EMC Corporation 2007
📖 EMC ControlCenter Online Help V5.2