

CiscoWorks Resource Manager Essentials Compliance Management

Introduction	1
Technical details	2
Working with Baseline Templates.....	2
Creating a Baseline Template	2
Running a Compliance Check	4
Interpreting Baseline Compliance Reports	5
Deploying a Template.....	6
Editing a Baseline Template	7
Exporting Baseline Templates.....	7
Importing a Baseline Template	8
Working with Advanced Templates	9
Limitations of Compliance Management with Baseline Templates.....	27
References	28

Introduction

In today's world, with complicated networks spread across continents, every action that happens or does not happen to a network can affect its state of compliance.

Consider this example: While deploying a new device, the administrator fails to change the configuration template's Simple Network Management Protocol (SNMP) community string settings. Because the template might be accessible to a number of people, the default SNMP community string is likely to be known by everyone who has access. Anyone with this information can reconfigure portions of the new device, thus compromising the security of the device. Once compromised, the device can become a gateway for access to data from the network or a means to introduce malicious software in the network without anyone's awareness. A seemingly innocent mistake, the failure to change a text string consisting of a dozen characters or so could result in an entire organization accidentally disclosing sensitive data with no accountability.

Even a small, incorrect modification to the access control list (ACL) in a router can lead to a serious security breach and result in unauthorized data access.

Thus, network compliance is of great significance to any organization. To help ensure network compliance and thereby organizational data security and integrity, there is a need to have a tool to monitor network compliance and deploy the required changes in the network to make it compliant.

CiscoWorks Resource Manager Essentials (RME), part of the CiscoWorks LAN Management Solution (LMS), is a powerful set of web-based application features that offer enterprise-class network management of Cisco® switches, access servers, and routers. CiscoWorks RME allows easy access to information that is critical to network uptime and simplifies time-consuming administrative tasks.

Technical details

Working with Baseline Templates

A baseline template contains placeholders for device-specific values to be substituted, as described in the following example:

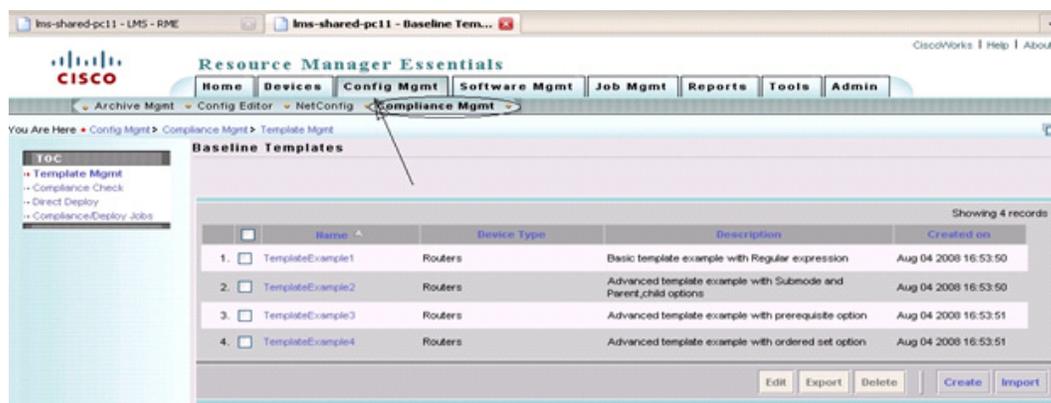
```
+ snmp-server community [Value] RW
```

Where + means that the command is expected to be present in the configuration, and Value is the community string that needs to be set. When a new device of the same type is added to the network, the existing baseline template (like the one in the preceding example, consisting of two parts, command and values) can be used. Configurations for any device of the same type can be created by specifying the values for the variables in the baseline template once. The following sections will use examples to describe and explain how to create and use a baseline template.

Example: In a complicated and wide-spanning network, the network administrator needs to check whether the community strings configured in the devices comply with the organizational standards or not.

To perform this task, the network administrator needs to create a baseline template first (refer to the “Creating a Baseline Template” section for details). Suppose that all the community strings should adhere to a standard that states that the read community strings should all start with “Cisco” and the write community strings should all start with “Sisco”.

Figure 1. Checking Community String Configuration of a Device

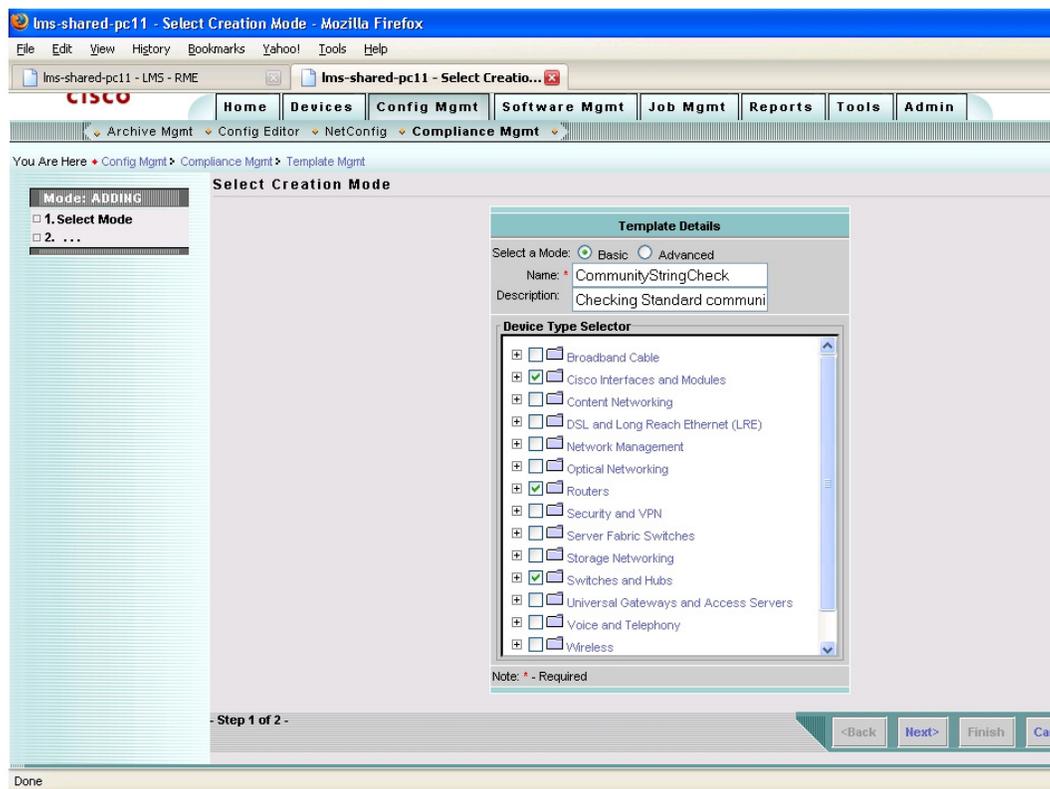


Creating a Baseline Template

A baseline template needs to be created to run a compliance check for configuration of various devices in the network. The network administrator can run a check on the devices in the network, check their compliance, and generate a report using this template.

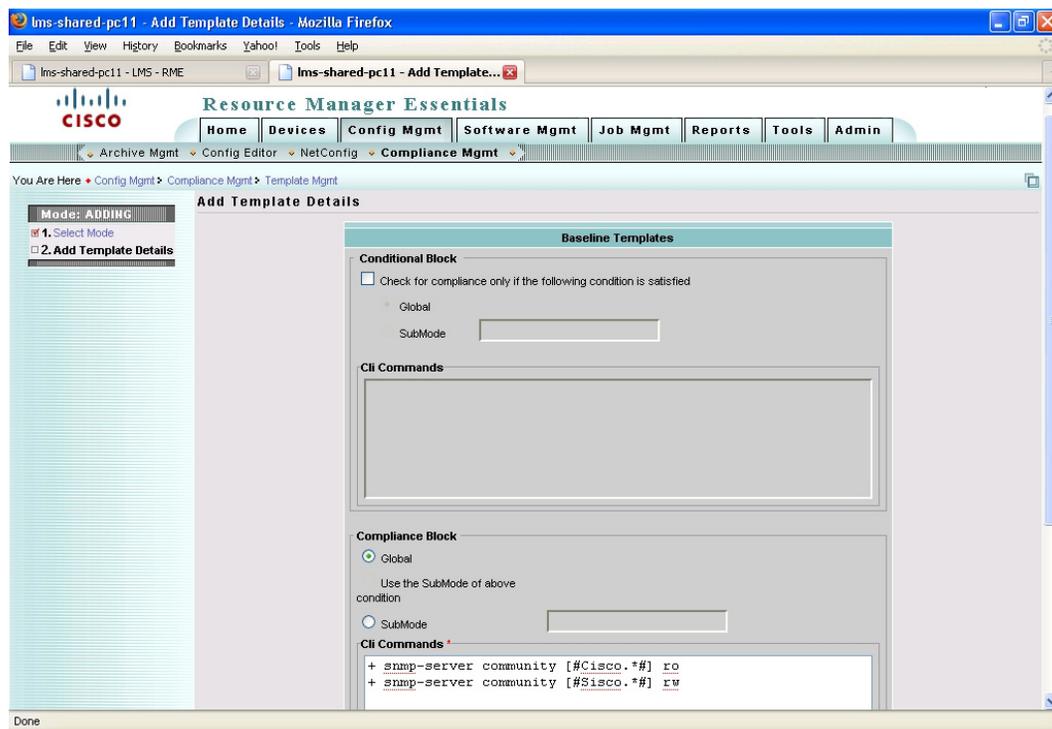
To create a baseline template:

1. Choose **Resource Manager Essentials >> Config Mgmt >> Compliance Mgmt >> Template Mgmt >> Create** (refer to Figure 1). The Select Creation Mode screen appears (refer to Figure 2).

Figure 2. Creating a Baseline Template

2. Select the appropriate device category, click **Basic**, and provide an appropriate description and a name for the template in the text area. (For advanced template creation, refer to the “Working with Advanced Templates” section later in the document.) The Add Template Details screen appears (refer to Figure 3).

Figure 3. Adding Template Details



3. In the Baseline Templates window, key in the policy-based commands (as desired in the device configuration) for the compliance check and click **Finish**. The template is created and can now be chosen for a compliance check job.

Example: To check the community strings, configured according to the standards (that is, starting with Cisco for the read and Sisco for the write community), the commands should be:

```
+ snmp-server community [#Cisco.*#] ro
+ snmp-server community [#Sisco.*#] rw
```

Running a Compliance Check

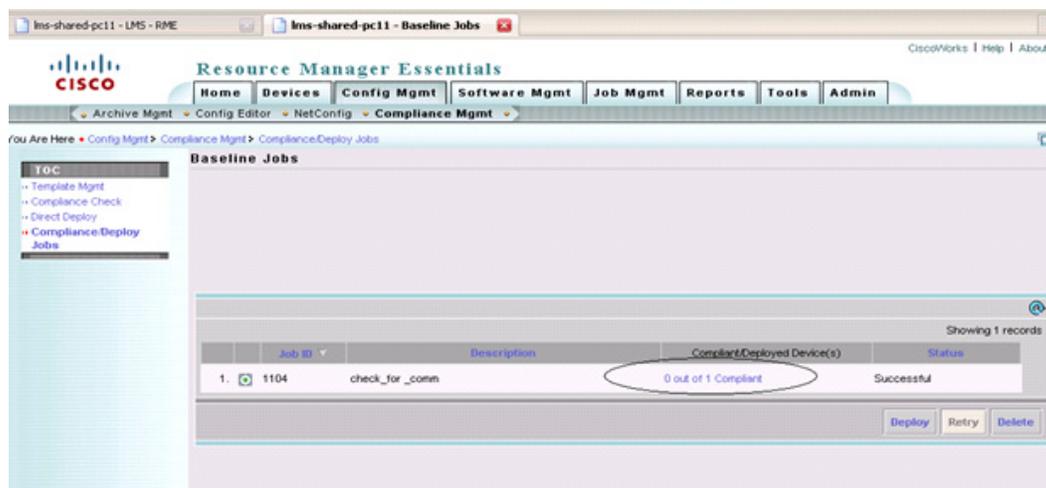
After a baseline template is created, the CiscoWorks RME Baseline Template application is ready to run a compliance check on all the identified devices in the network.

To run a compliance check:

1. Choose **Resource Manager Essentials >> Config Mgmt >> Compliance Mgmt >> Compliance Check**, select the appropriate template you want to run a compliance job, and click **Compliance Check**.
2. Select the appropriate devices from Device Selector or groups from Group Selector that need to undergo the compliance check and click **Next**. The job schedule screen appears. Select immediate from Run Type and provide a name for the job and click **Finish**. Go to the **Compliance/Deploy Jobs** tab to see the status of the job. The Baseline Jobs screen appears (refer to Figure 4).

Note: The total number of devices chosen in the job and the number of compliant devices will be shown in the job browser window (refer to Figure 4). The compliance job reports the number of devices that are compliant (out of the total number of devices chosen in the job). Clicking the hyperlink (0 out of 23 Compliant), as shown in Figure 4, provides more information on the noncompliance, which includes the devices that are noncompliant, the commands that are noncompliant, and so on.

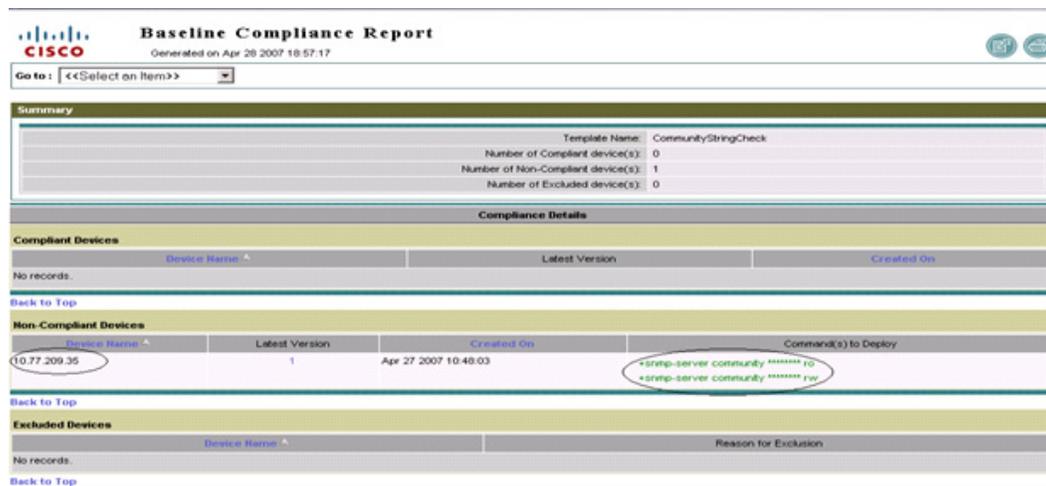
Figure 4. Running a Compliance Check



Interpreting Baseline Compliance Reports

The compliance reports are designed based on usability principles to help ensure ease of use and understanding.

Figure 5. Understanding a Sample Compliance Report

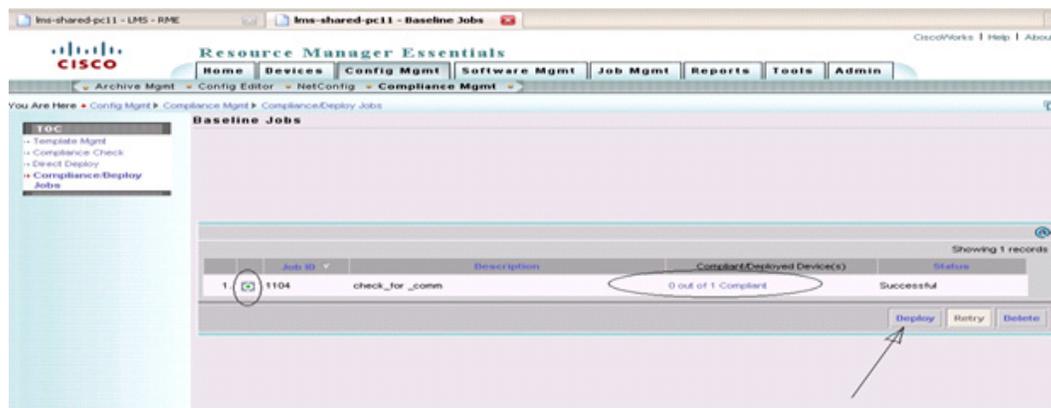


Noncompliance or commands not available in the device configuration for respective devices are highlighted in green. The commands that are expected to be present (that is, starting with +) are shown in green, and those that are not expected to be present (that is, starting with -) are shown in red in the compliance report. Refer to Figure 5 for a better understanding.

Deploying a Template

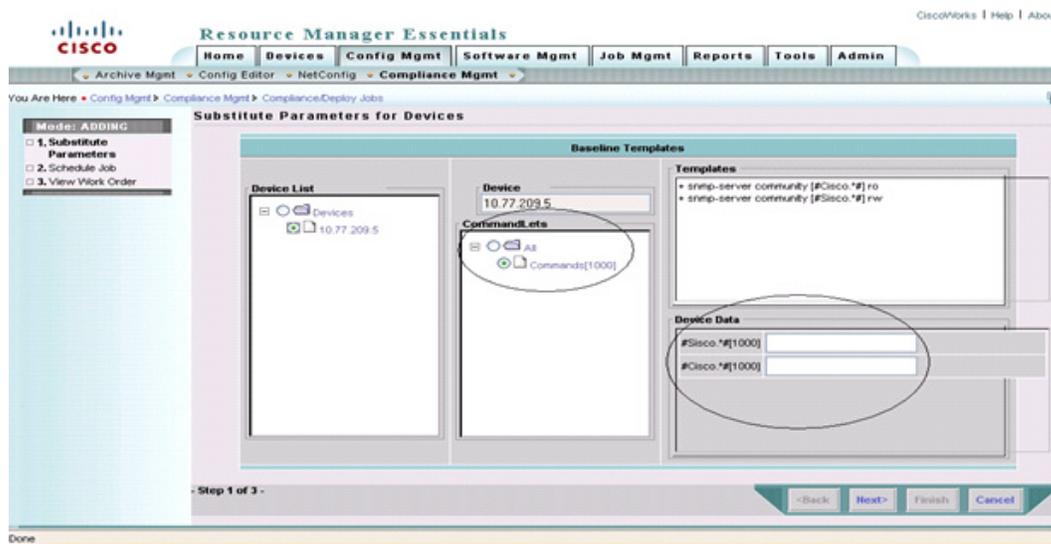
The Baseline Template application of CiscoWorks RME makes it possible to deploy a baseline template, use it for a compliance check on a noncompliant device, and make the device compliant. In the Community String compliance example cited above, it was observed that one device was noncompliant. To make the device compliant, use the following steps:

Figure 6. Deploying a Compliance Template



1. In the Baseline Jobs screen (refer to Figure 6), select the appropriate compliance job ID you want to deploy and click **Deploy**. The Substitute Parameters for Devices screen appears (refer to Figure 7).
2. Provide the command values that need to be deployed on the device. Click **Next** and schedule the job.

Figure 7. Deploying a Template



Note: For example, in Figure 7, clicking **Commands [1000]** under CommandLets will populate the field names in the Device Data section. Provide appropriate values in the Device Data section and schedule the job. The template will be deployed on the required device. The status of the deployed job can be checked by choosing **Config Mgmt >> Compliance Mgmt >> Compliance/Deploy Jobs**.

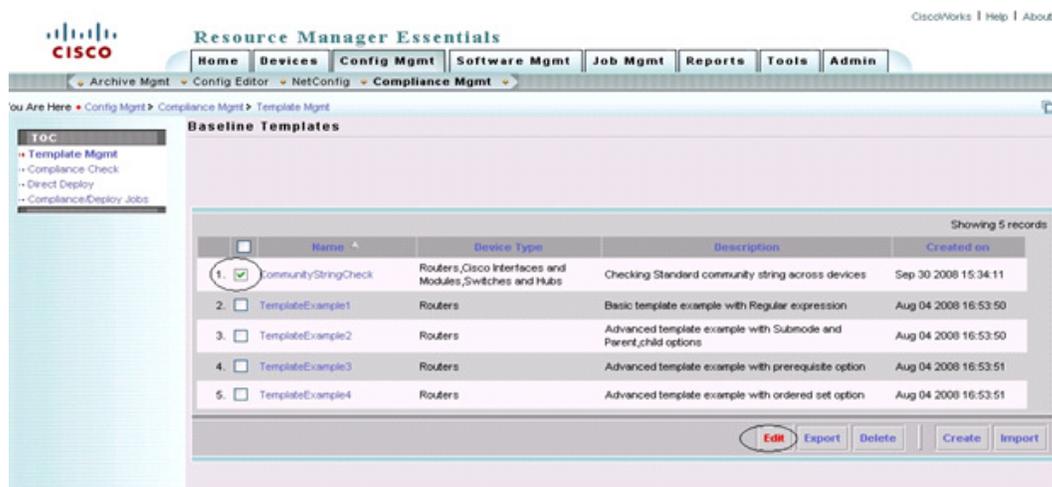
In the Community String example, the read community should start with “Cisco” and the write community should start with “Sisco”. Providing the appropriate community strings in the Device Data section and scheduling the job would deploy the changes on the device.

Editing a Baseline Template

An existing baseline template can be edited and customized to make it suitable for specific networks. To edit a baseline template:

1. Choose **Config Mgmt >> Compliance Mgmt >> Template Mgmt**. The Baseline Templates screen appears (see Figure 8).

Figure 8. Editing a Baseline Template



2. Select the template you want to edit and click **Edit**. The Edit screen appears.
3. Make the changes in the Edit screen according to your requirement and click **Save**. The selected template is changed.

Note: While editing the template, the following parameters cannot be changed:

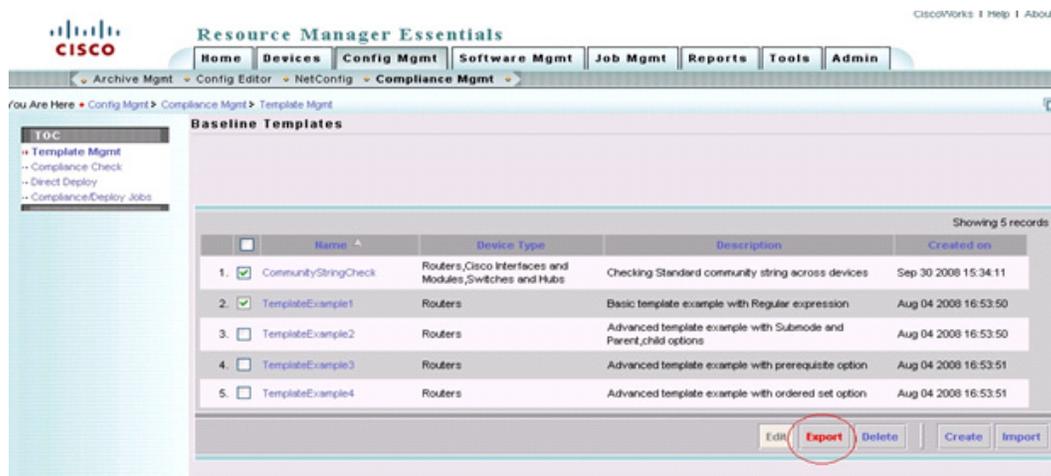
- Name of the template
- Template mode (Basic or Advanced)

Exporting Baseline Templates

CiscoWorks RME baseline templates created in one server can be exported as XML files and stored in the file system using the Export option. The exported templates can be reused by importing them in any other servers. To export a baseline template:

1. Choose **Config Mgmt >> Compliance Mgmt >> Template Management**. The **Baseline Templates** screen appears (refer to Figure 9).

Figure 9. Exporting Baseline Templates



2. Select the appropriate templates to be exported and click **Export**.

Note:

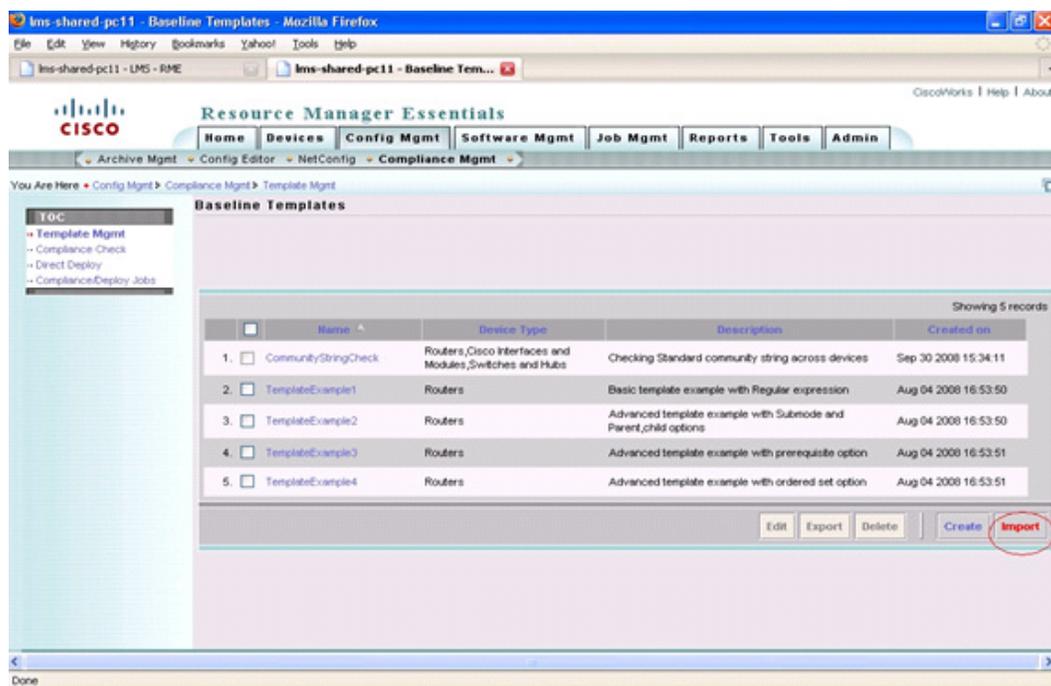
- Multiple templates can be exported at the same time.
- The templates can be exported only to a particular location as follows:
 - **Windows:** NMSROOT\files\rme\dcma\baselinetemplates
 - **Solaris:** /var/adm/CSCOpX/files/rme/dcma/baselinetemplates

Importing a Baseline Template

The exported baseline templates in XML format can be imported in other servers for reuse. To import a baseline template:

1. Choose **Config Mgmt >> Compliance Mgmt >> Template Mgmt**. The Baseline Templates screen appears (refer to Figure 10).

Figure 10. Importing a Baseline Template



- Click **Import** and select the appropriate templates to be imported from the new window that appears.

Note:

- After importing the template, it will be listed in the Baseline Templates page.
- The templates can be imported only from the following locations:
 - Windows:** NMSROOT\files\rme\dcma\baselinetemplates
 - Solaris:** /var/adm/CSCOpX/files/rme/dcma/baselinetemplates

Working with Advanced Templates

This document has discussed working with basic templates and how they can be used to help ensure network compliance. In the following sections, the document discusses advanced templates and how to create and work with them.

Creating and Using an Advanced Baseline Template

The basic steps for creating an advanced baseline template are the same as those for creating a basic template, the only difference being in choosing Advanced instead of Basic in the template creation mode (refer to Figure 2). Before starting to work with advanced baseline templates, it is important to have a clear understanding of the concept of different types of command sets.

Command Set

As discussed earlier, this is a set of one or more command-line interface (CLI) commands. The basic characteristics of a command set are:

- Each command set can have one or more child command sets.

- The child command set inherits a parent's submode command. If there is no submode command mentioned in the child command set, it uses the parent command set's submode command.
- If the commands in a command set are in a submode (IP/interface, and so on), a submode command must be specified for such a command set. Otherwise, the baseline template would look for the commands in the wrong location (for example, in the global configuration) thereby leading to unexpected results.

Prerequisite Command Set

A command set can be marked as a prerequisite for another command set. For example, assume there are two command sets:

- **Command Set A:** This is marked as a prerequisite command set of Command Set B.
- **Command Set B:** B has A as its prerequisite.

A prerequisite command set is used only for comparison and is not deployed on any device. In the example, Command Set A is a prerequisite of Command Set B. When CiscoWorks RME evaluates the baseline template, Command Set A is evaluated first and B is evaluated next. If A's evaluation leads to false, Command Set B is not evaluated at all. The device is considered noncompliant and will be shown in the excluded device list in the compliance report.

Checking for Compliance Using Prerequisites

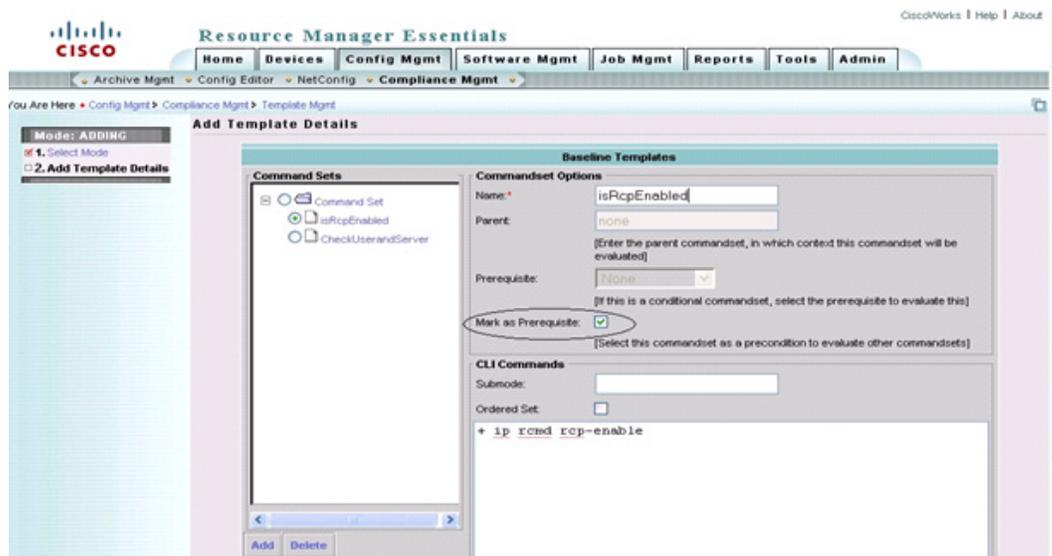
For the CiscoWorks RME configuration archive to gather configurations using the Remote Copy Protocol (RCP), devices in the network need to be RCP-enabled. Quite often, when RCP is involved, the configuration jobs fail with a message stating, "Configuration archive failed. Check if RCP is enabled on your device." Baseline templates can be used to check whether the devices on the network are RCP-enabled or not. In the following example, this compliance check will be performed using prerequisite command sets. Apply the two command sets as discussed in the preceding example.

Command set *IsRcpEnabled* will check whether the server side of RCP is enabled in the device. This command set will be marked as a prerequisite command set.

To perform this configuration check:

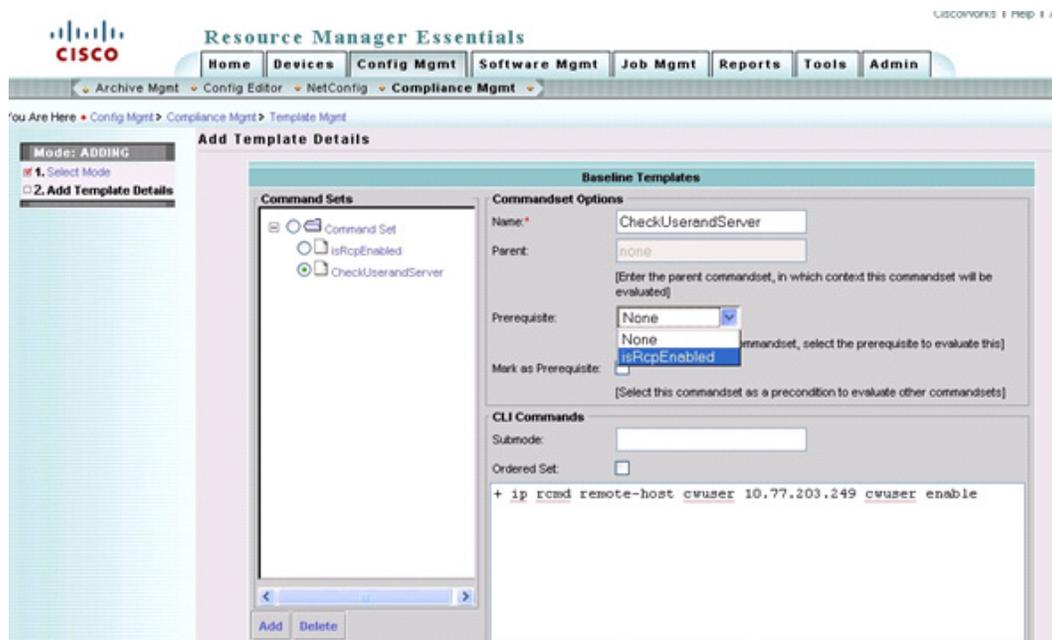
1. Choose **Config Mgmt >> Compliance Mgmt >> Template Mgmt** and click **Create** to create a baseline template.
2. Choose **Advanced** to create an advanced template and provide a name and appropriate description for the template.

Figure 11. Using Prerequisite Template Details



3. Select the device categories where the template needs to be applied and click **Next**.
4. Select **Global Command Set** in the Add Template Details window.
5. Provide the name of the template as **IsRcpEnabled**, and select **Mark as Prerequisite** (refer to Figure 11).
6. Provide the CLI commands in the text area and click Save. A confirmation dialog box is displayed.
7. Add and select **IsRcpEnabled** as the prerequisite of CheckUserAndServer. A new command set for CheckUserAndServer is created (refer to Figure 12).

Figure 12. Working with Prerequisites



Note: For CiscoWorks, cwuser is the default username. Assuming this, a typical configuration would be like *ip rcmd remote-host cwuser hostname_or_ip cwuser enable*.

8. Click **Save** and then click **Finish**. The template with the new configuration is created.

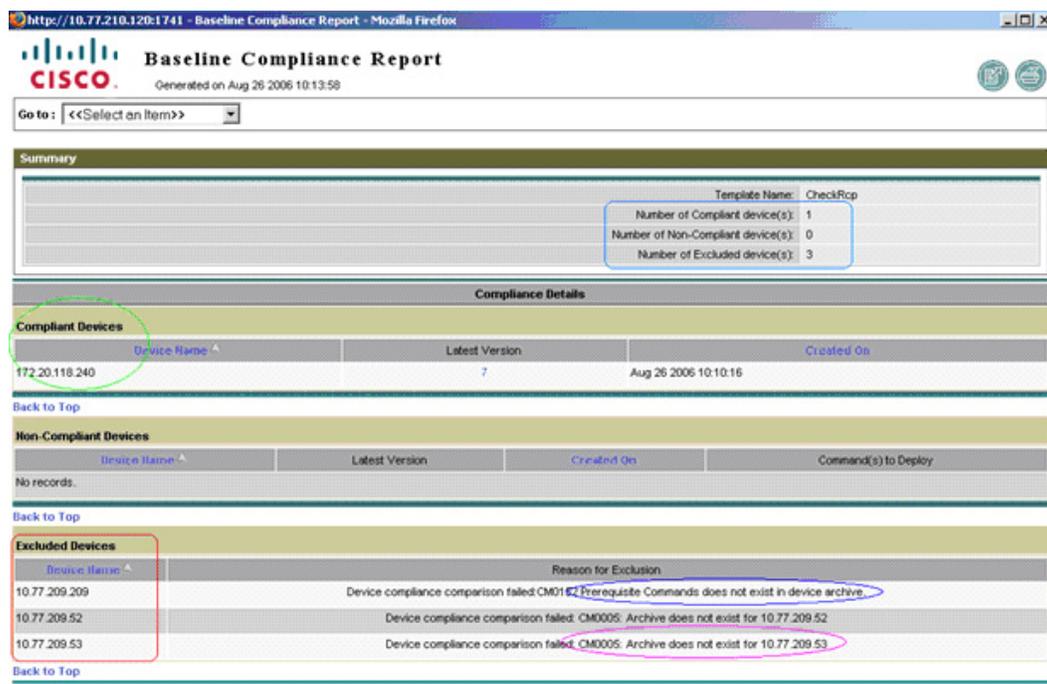
Note: If the command sets consist of a prerequisite command set, these command sets appear in red color in the Preview details. Clicking **Preview** will allow the network administrator to see the prerequisite commands in red.

If required, wildcard characters can also be used in advanced templates. But in the preceding example, since the objective is to look for RCP configuration for the CiscoWorks server, this may not be required. Once the new template is created, running a compliance job will help enable the network administrator to know whether any of the network devices have RCP enabled for the CiscoWorks server or not.

How a Baseline Template Works

This section uses an example of prerequisite command sets to explain how baseline templates work. As discussed earlier, the IsRcpEnabled command set is evaluated first. The rules for this command set are that the *ip rcmd rcp-enable* command should be present in the configuration. This can lead to two results:

- If this command is present in the device configuration, the command set CheckUserAndServer will be evaluated, which will check for the presence of the *ip rcmd remote-host cwuser 10.77.203.249 cwuser enable* command in the configuration. If CheckUserAndServer evaluates to true, then the device is compliant; otherwise, it is noncompliant.
- If IsRcpEnabled evaluates to false, there is no *ip rcmd rcp-enable* command in the configuration. This signifies that the server-side RCP is not enabled and there is no point in proceeding further. Thus, CheckUserAndServer will not be evaluated at all, and the corresponding device will be marked as an excluded device as shown in Figure 13.

Figure 13. Understanding a Baseline Compliance Report

In the preceding example (refer to Figure 13), only one out of four devices was compliant. The reason behind the other three devices being categorized under excluded devices is circled in blue and pink. One device was noncompliant due to the failure of the prerequisite command set, and the other two failed because they did not have any configuration in the archive. This example works well only for compliance. If you want to deploy the changes, if any of these command sets fails, then a parent/child command set will be more appropriate. This is described later in the document.

Excluded and Noncompliant Devices

In the report shown in Figure 13, there are three devices in the excluded category. They are different from noncompliant devices. If the compliance job for some devices did not run or could not proceed due to some reason, the devices are categorized as excluded. The reasons for a device being categorized in the excluded list are:

- Device configuration was not archived
- Prerequisite command set failed

Figure 13 depicts both these cases.

While noncompliant devices can be made compliant, by deploying the changes using a baseline template, the same does not hold true for an excluded device. In the case of a prerequisite failure, the device is marked as excluded instead of noncompliant because the prerequisite command set is considered only for comparison/compliance and cannot be deployed. Thus, in such cases, a baseline template cannot be used to deploy the required changes and make the device compliant.

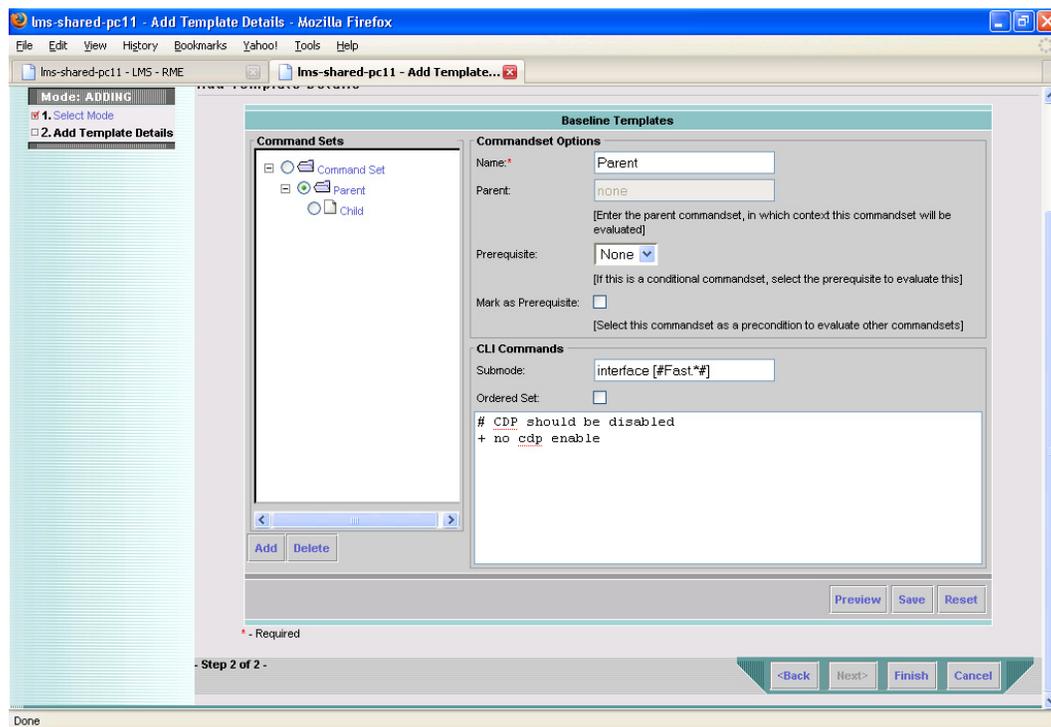
Parent and Child Command Sets

A command set can have one or more child command sets. A child command set inherits the parent's submode command. CiscoWorks RME evaluates the command sets in different ways depending on whether you have defined the command set as parent or prerequisite. The earlier example illustrated how CiscoWorks RME evaluates a prerequisite command set. In the case of a parent command set, it is slightly different. The following example illustrates this.

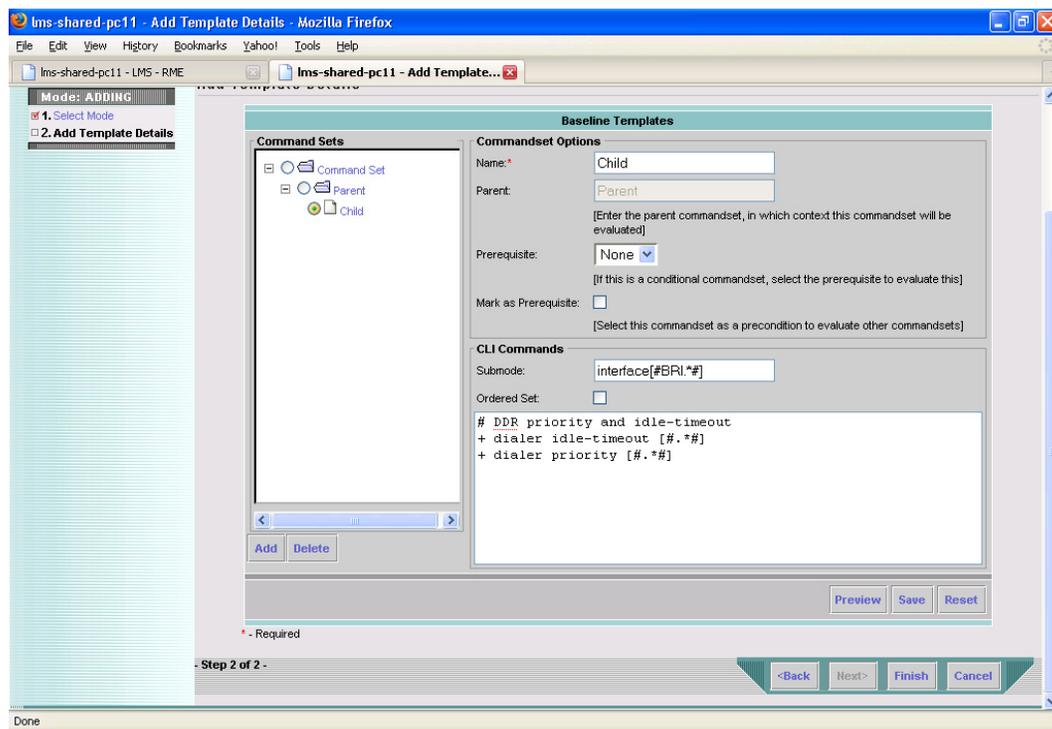
Example: Assume that Command Set A is the parent of Command Set B. While CiscoWorks RME evaluates the baseline template, Command Set A is evaluated first and Command Set B is evaluated next.

If either of these command sets is missing, the template is considered noncompliant. This is where the difference from a prerequisite is. In the earlier example, if the prerequisite fails, then the other command set is not even evaluated, and the device is marked as excluded. But in the case of a parent command set, if either A or B is false, the device is marked as noncompliant, which means that the baseline template (Figure 14) can be used to deploy the missing commands in the device configuration. The same example (checking RCP configuration) can be rewritten using a parent/child relationship, but this time, if either of the command sets is missing, the device will be marked as noncompliant, which means you can deploy the changes (missing command sets) on the device.

Figure 14. Using Command Sets



Parent/child command sets are used to logically group different command sets. Suppose you want to scan through all the interfaces and disable Cisco Discovery Protocol in all Fast Ethernet interfaces and additionally check for dial-on-demand routing (DDR) priority and idle-timeout on all Basic Rate Interfaces (BRIs). You can logically group Fast Ethernet-related tasks, say as a parent command set, and the BRI-related tasks as a child command set. If either of these command sets is false, the corresponding device is considered noncompliant.

Figure 15. Working with Child/Parent Command Sets

Note: While creating the child command set, the parent command set against the Parent field (shown in Figure 15) should always be mentioned.

Running a Compliance Job

A compliance job can be run by selecting the job ID and scheduling the changes to the devices (in case of noncompliance). To run a compliance job:

1. Invoke the compliance report by choosing **Config Mgmt >> Compliance Mgmt >> Compliance/Deploy Jobs**. The Baseline Jobs screen appears.
2. Select the appropriate compliance job and select the link appearing under Compliant/Deployed Device(s).

A compliance report with noncompliant devices will appear. The next step would be to review the compliance report and check how to deploy the noncompliant commands to the devices. The commands shown in green indicate that they are not present in the device, which is the reason for noncompliance. In the given report, Cisco Discovery Protocol is enabled in some of the Fast Ethernet interfaces and hence is noncompliant. To deploy the changes, select the appropriate compliance job ID and click **Deploy**.

Refer to Figure 16 for a sample screenshot of the compliance report.

Figure 16. Understanding a Baseline Compliance Report

http://10.77.240.86:1741 - Baseline Compliance Report - Mozilla Firefox

Baseline Compliance Report
Generated on -

Go to: <<Select an Item>>

Summary

Template Name:	ParentChild
Number of Compliant device(s):	0
Number of Non-Compliant device(s):	1
Number of Excluded device(s):	0

Compliance Details

Compliant Devices

Device Name ^	Latest Version	Created On
No records.		

[Back to Top](#)

Non-Compliant Devices

Device Name ^	Latest Version	Created On	Command(s) to Deploy
10.77.209.35	1	Apr 27 2007 10:48:03	+interface FastEthernet3/1 + no cdp enable +interface FastEthernet3/5 + no cdp enable +interface FastEthernet3/7 + no cdp enable +interface FastEthernet3/12 + no cdp enable

[Back to Top](#)

Excluded Devices

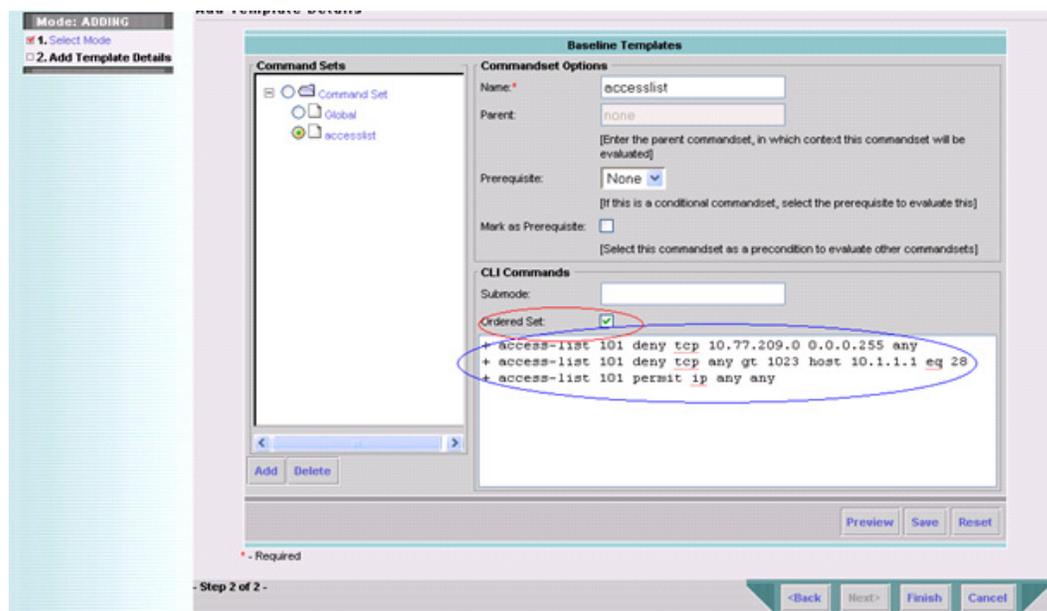
Device Name ^	Reason for Exclusion
No records.	

Working with Ordered Sets

The order of the commands in a device configuration is significant at times. A typical example is an ACL. Since the access list rules are evaluated in the order in which they appear (sequential), it is very important that the order of ACLs is correct. While creating an advanced baseline template, you have an option to inform CiscoWorks RME that the commands in the device configuration should appear in the same order as that of the CLI commands definition order in the command set. To help ensure that the order of commands is correct:

1. Choose **Config Mgmt >> Compliance Mgmt >> Templates Mgmt** and select the **Ordered Set** checkbox (Figure 17).

Figure 17. Ordering the Commands



2. Key in the commands in the preferred order in the text box. This will help ensure that during the compliance check the commands run in the preferred order.

Deployment of a Baseline Template

Deploying a baseline template can be done in two ways:

- Run a compliance check first and then, based on the compliance report, deploy the changes on the noncompliant devices.
- Deploy a baseline template directly, either by using the user interface or through the file system.

The following sections describe these two approaches.

Deploying a Baseline Template through a User Interface

To deploy a baseline template:

1. Choose **Config Mgmt >> Compliance Mgmt >> Template Mgmt**. The Baseline Templates dialog box appears.
2. Select a baseline template and click **Deploy**. The Deploy Input Options dialog box appears.
3. Select **Enter Data From User Interface** and click **Next**. The Select Devices dialog box appears.

Note: The device list contains only those particular device type devices that you have selected while creating the baseline template. For example, if you have selected the device type as Router, then only routers are listed.

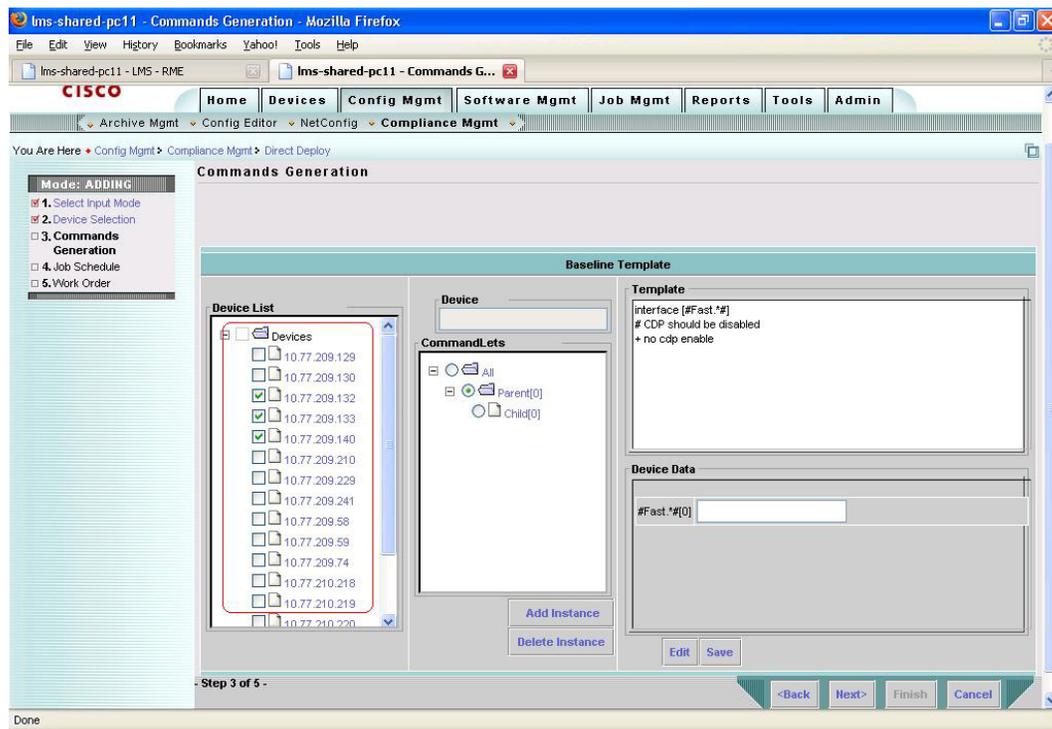
Devices are grouped in the following tabs:

- In the All tab, devices are grouped under All Applicable Devices and All Applicable Device Groups. All Applicable Device Groups categorizes devices under routers, switches, and so on.
- In the Search Results tab, the results of simple search and advanced search are listed.

- In the Selection tab, all devices that are selected are listed. The devices can be deselected in this tab.

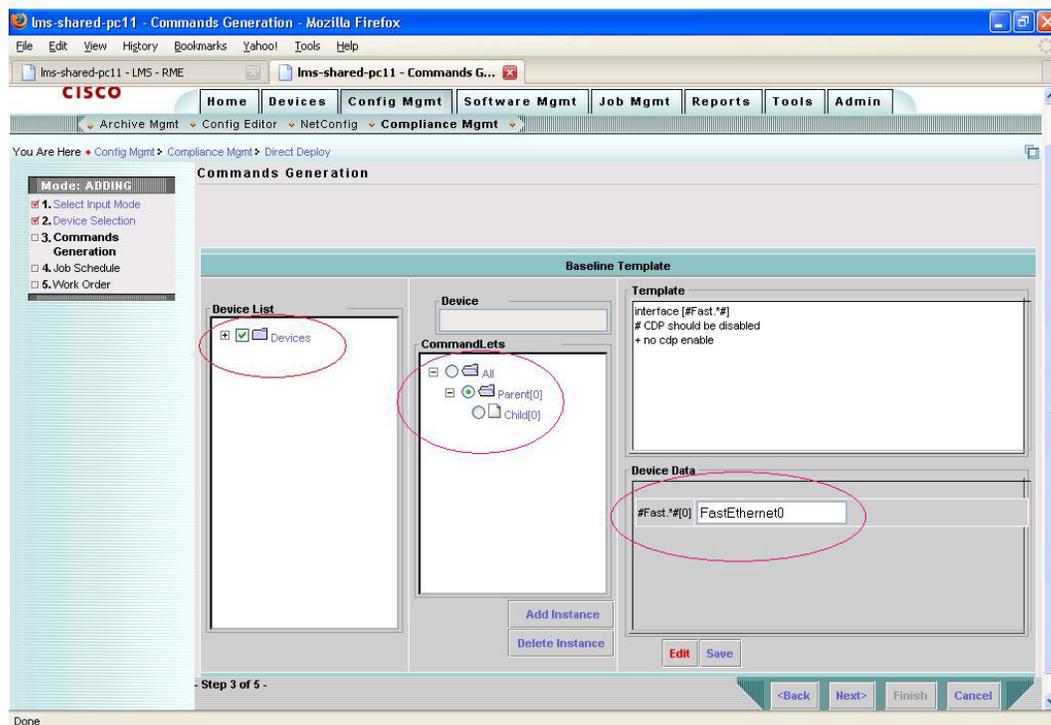
For ease of understanding, consider the parent/child template developed in the earlier section to provide values using the user interface. Refer to Figure 18 for a better understanding.

Figure 18. Deploying a Baseline Template through the User Interface



In Figure 18, the Device List pane shows the list of devices that have been chosen for deployment. The next pane shows the command sets in the chosen template; in this case it is the parent and child command sets. The Templates area shows you details of the chosen command set. Data can be provided for all the devices at one stretch. In the example, the parent command set is supposed to disable Cisco Discovery Protocol in all the Fast Ethernet interfaces and the child command set is supposed to set the DDR priority and idle timeout on all BRIs. All the devices from the Device List pane can be selected and values provided for parent and child command sets. These values, once set, will be applied to all devices. To change values for a subset of devices, those devices can be selected individually and values provided for the selected devices alone. Once the data is provided, clicking Save will save the device data and proceed to finish the job. An information dialog box to inform that the device data is saved successfully will pop up. Refer to Figure 19 for details.

Figure 19. Working with Device Lists



All devices are chosen and the values for the parent command set provided. Similarly, values are provided for the child command set as well.

To edit the values for any one of the devices:

1. Select the device from the Device List pane and click **Edit**. The Device Data section appears.
2. Provide the values for the specified device and click **Save**. The values for the device are updated (refer to Figure 20).

Figure 20. Editing Values for a Specific Device

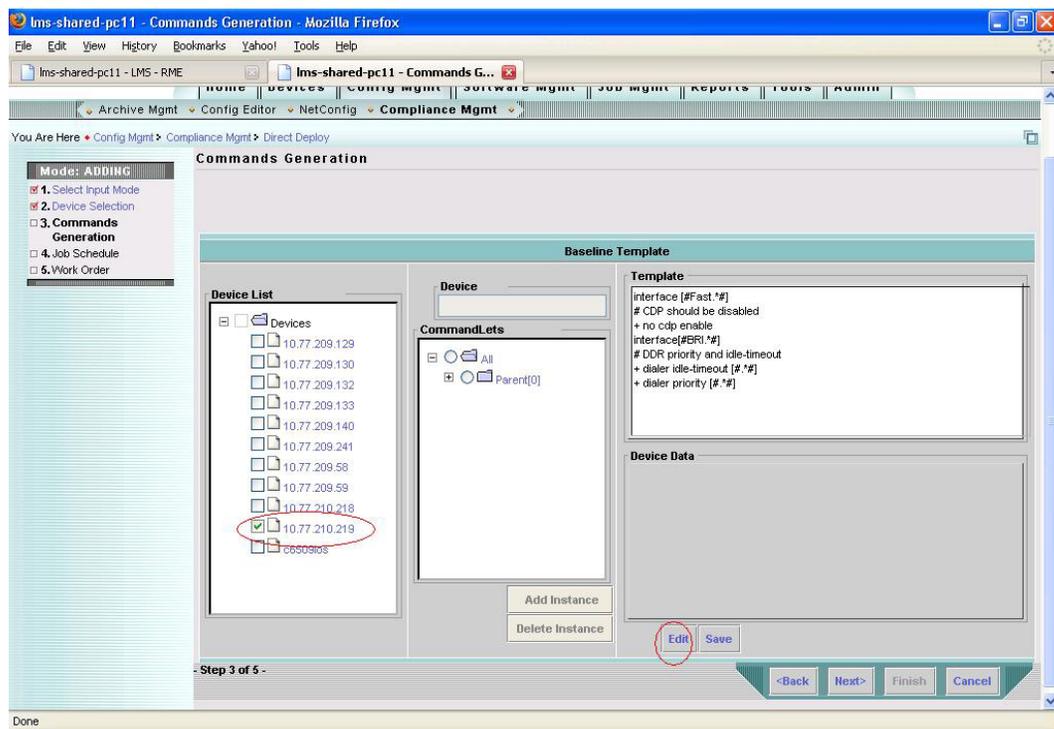
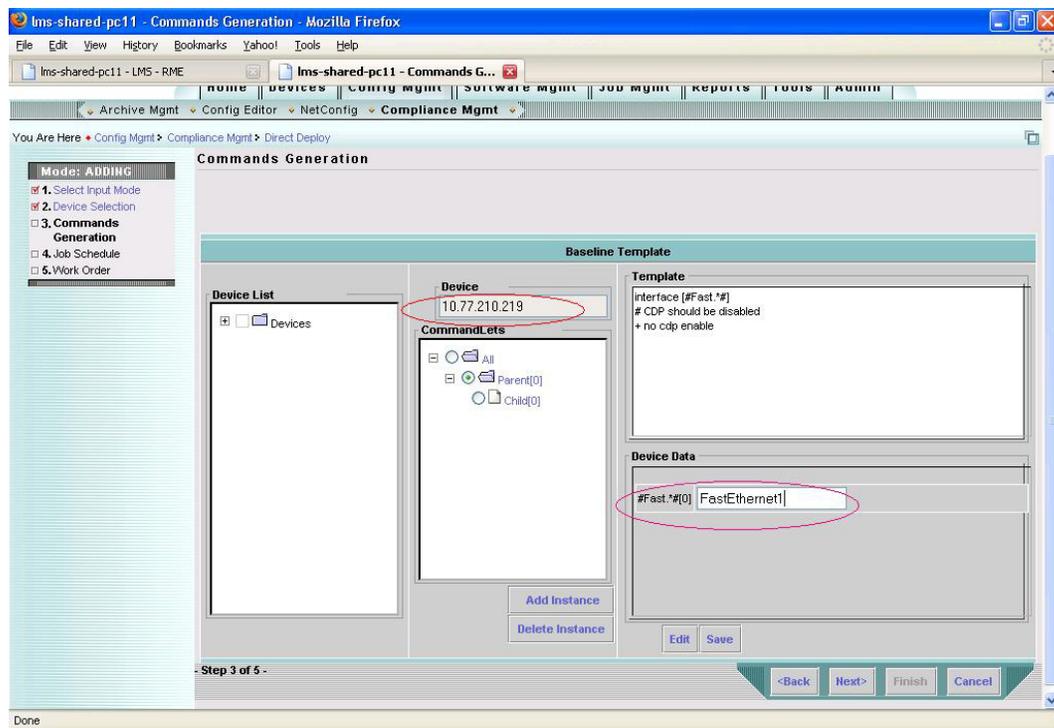


Figure 21. Generating Commands



3. Click **Next** and then **Finish** to complete the task (refer to Figure 21).

Note: Clicking the **Add Instance** button will add one more instance of a multiple command set in the Command Sets pane. This is useful when a command set appears more than once. An example could be an access list, which is present many times in a device configuration.

Parameterized Config Deployment using XML

You can use the baseline parameter file to deploy the baseline template on CiscoWorks RME devices. While command deploy is possible using other modules in RME what makes deployment in compliance management with baseline templates more powerful is that this feature allows users to use regular expressions for deploying commands on to the devices. To do this, use the following steps:

1. Choose **Config Mgmt >> Compliance Mgmt >> Template Mgmt**.
2. Select Enter Data from File System.

Generating a Baseline Parameter File

To generate the baseline parameter file in XML format:

1. Select **Config Mgmt >> Compliance Mgmt >> Template Mgmt**. The Baseline Templates dialog box appears.
2. Select a baseline template and click **View**. The Baseline Config Viewer window appears.
3. Click Generate Param File and:
 - Either enter the directory name on the CiscoWorks server to save the baseline parameter file, or
 - Click **Browse**.

The Server Side File Browser dialog box appears.

4. Select a directory on the CiscoWorks server to save the baseline parameter file and Click **OK**.

A message, Parameter file generated as: \User_Specified_Directory\Template Name-param.xml, appears.

Note: The naming convention followed for the baseline parameter file is Template Nameparam.xml.

Understanding the Baseline Parameter File

The baseline parameter file consists of the following sections:

Device Hostname: This contains the device display name as entered in the Device and Credential Repository. You can specify the device display name using the syntax:

```
<Device HostName="1.1.1.1">
</Device>
```

You must specify at least one value for device hostname.

DataSet CommandLet Name: The command set name entered during baseline template creation.

Name: The command variable name as entered while creating the baseline template.

Example: + banner motd [message]

Here, the name of the variable would be message.

Value: Enter the value for the command variable.

```
<Value>Command_Variable_Value</Value>
```

The following is a sample parameter template file for the parent/child example discussed earlier in this document:

```
<?xml version="1.0" encoding="UTF-8"?>
<ConfigData>
<Device HostName="10.77.209.246">
  <DataSet CommandletName="Parent">
<Param>
<Name>#Fast.*#</Name>
<Value>FastEthernet0</Value>
</Param>
  <DataSet CommandletName="Child">
<Param>
<Name>#.*#</Name>
<Value>10</Value>
</Param>
<Param>
<Name>#BRI.*#</Name>
<Value>BRI1/0</Value>
</Param>
  </DataSet></DataSet>
  </Device>
<Device HostName="10.77.209.35">
<DataSet CommandletName="Parent">
<Param>
<Name>#Fast.*#</Name>
<Value>FastEthernet0</Value>
</Param>
<DataSet CommandletName="Child">
<Param>
<Name>#.*#</Name>
<Value>10</Value>
</Param>
<Param>
<Name>#BRI.*#</Name>
<Value>BRI1/0</Value>
</Param>
  </DataSet></DataSet>
  </Device>
  <Device HostName="172.20.118.177">
<DataSet CommandletName="Parent">
<Param>
<Name>#Fast.*#</Name>
<Value>FastEthernet1</Value>
</Param>
<DataSet CommandletName="Child">
```

```

<Param>
<Name>#. *#</Name>
<Value>10</Value>
                                     </Param>

<Param>
<Name>#BRI.*#</Name>
<Value>BRI1/0</Value>
</Param>
      </DataSet></DataSet>
</Device>
</ConfigData>

```

Regular expressions can be very useful in text, string search, and manipulation. Baseline templates use regular expressions extensively, and together they make a powerful combination. Here is a quick overview of some of the literals and wildcards involved in regular expressions:

- **Dot operator (.):** A dot in a regular expression would match any single character.

Example: `gr.y` matches `gray`, `grey`, `gr%y`, and so on.

- **Anchors:** Anchors do not match any characters but match a position. `^` matches at the start of the string, and `$` matches at the end of the string.

Example:

`^b` matches `"bob"`, `"big"`, `"bat"`, and so on.
`b$` matches `"bob"`, `"tab"`, and so on.

- **Star operator (*):** A star in a regular expression would match zero or more occurrences of the previous character.

Example:

`A*B` would match `"AB"`, `"B"`, `"AAB"`, `"AAAAB"`, and so on.

- **Combination of dot and star:** This would match any character set.

Example:

`.*` would match `"Anything"`, `"Everything"`, and so on.
`A.*B` would match anything that starts with `A` and ends with `B`, including `"AB"`.

- **Plus operator (+):** This would match one or more occurrence of a previous character.

Example:

`A+B` would match `"AB"`, `"AAB"`, `"AAAAAB"`, and so on, but not `"B"`.

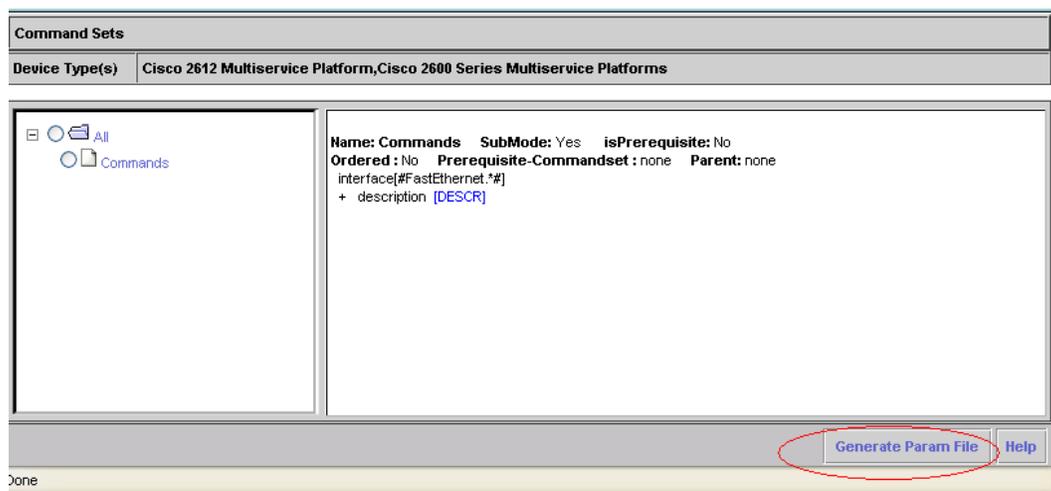
- **Question mark (?):** This would match either zero or one occurrence of the previous character.

Example:

`A?B` would match `"AB"` or `"B"` only.

How to Deploy Using the Parameterized File

For example if you want to check and deploy a description for each interface in a particular device, you must create a template of the following format and generate a parameterized file from the created template. This can be done from baseline template viewer window. You can open the baseline template viewer window by clicking on the link available in the template management page. Refer to Figure 22 Generating a Parameter File.

Figure 22. Generating a Parameter File

The generated parameterized file will be of the following format.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ConfigData>
<Device HostName="Default">
<DataSet CommandletName="Commands">
<Param>
<Name>#FastEthernet.*#</Name>
<Value />
</Param>
<Param>
<Name>DESCR</Name>
<Value />
</Param>
</DataSet>
</Device>
</ConfigData>
```

Users will have to edit the values tag for as many number of interfaces they want to edit and save the file in a file system. For example, the edited XML might look like that shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ConfigData>
<Device HostName="10.77.0.1">
<DataSet CommandletName="Commands">
<Param>
<Name>#FastEthernet.*#</Name>
<Value>FastEthernet0/0<Value/>
</Param>
<Param>
<Name>DESCR</Name>
<Value>Interface_uplink<Value/>
</Param>
</DataSet>
</Device>
```

```
<Device HostName="10.77.209.2">
  <DataSet CommandletName="Commands">
    <Param>
      <Name>#FastEthernet.*#</Name>
      <Value>FastEthernet0/0</Value/>
    </Param>
    <Param>
      <Name>DESCR</Name>
      <Value>Connecting_host</Value/>
    </Param>
  </DataSet>
</Device>
<Device HostName="10.77.209.1">
  <DataSet CommandletName="Commands">
    <Param>
      <Name>#FastEthernet.*#</Name>
      <Value>FastEthernet0/1</Value/>
    </Param>
    <Param>
      <Name>DESCR</Name>
      <Value>Connecting_mail_server</Value/>
    </Param>
  </DataSet>
</Device>
<Device HostName="10.77.209.2">
  <DataSet CommandletName="Commands">
    <Param>
      <Name>#FastEthernet.*#</Name>
      <Value>FastEthernet0/1</Value/>
    </Param>
    <Param>
      <Name>DESCR</Name>
      <Value>connecting_device_209.3</Value/>
    </Param>
  </DataSet>
</Device>
<Device HostName="10.77.209.2">
  <DataSet CommandletName="Commands">
    <Param>
      <Name>#FastEthernet.*#</Name>
      <Value>FastEthernet0/2</Value/>
    </Param>
    <Param>
      <Name>DESCR</Name>
      <Value>Connecting_gateway</Value/>
    </Param>
  </DataSet>
</Device>
```

```
</ConfigData>
```

Select the template from the table in Direct deploy page, and select option **Enter Data** From File System. Refer to Figure 23 **Selecting File System Option**.

Figure 23. Selecting File System Option

The screenshot shows a dialog box titled "Select Input Mode". It contains two radio button options: "Enter Data From User Interface" (which is selected) and "Enter Data From File System".

Select the parameterized file from the location and click **Finish** to schedule a job to do a direct deployment of these commands on the devices. Refer to Figure 24.

Figure 24. Scheduling a Job for Direct Deployment

The screenshot shows the "Job Schedule" configuration page. It includes the following sections and fields:

- Scheduling:** Run Type: Once; Date: 03 Mar 2009; Time: 17:10.
- Job Info:** Job Description: Description_deploy; E-mail: abc@cisco.com.
- Job Options:**
 - Approval Comment: This is to add description to the interfaces
 - Maker E-Mail: john@cisco.com
 - Copy Running Config to Startup
 - Job Password
 - Login user name: [text box]; Password: [text box]
 - Enable Password: [text box]

* - Required

This process negates the user interventions with the application for providing the data necessary for deploying the commands. This deployment method is more useful for users who are more comfortable with deployment using the CLI than using the UI.

Rules for Specifying a Baseline Template

While creating a baseline template, the following syntactical rules need to be followed:

- All mandatory configuration commands should begin with a +.
- All disallowed configuration commands should begin with a –.
- All commands that do not begin with – or + are considered as comments and are ignored.
- All commands should have a space between the – or +. If there is no space, the commands are considered as comments and are ignored.
- Anything between [# #] is considered as a regular expression and can have wildcard characters as explained earlier

```
+ ip address [#10\.76\.38\..*#] [netmask]
+ ip address [#10\.72\..*\..*#] [netmask]
Submode: interface [#Ethernet.*#]
Submode: interface [#Serial.*#]
Submode: interface [#Vlan.*#]
Submode: Interface [#Vlan.*|Ethernet.*#]
```

If you want to check for a specific configuration on all interfaces except Loopback, you can provide the submode command in the following format:

```
Submode: interface [#!Loopback.*#]
```

To find a match for any octet in an IP address, you must use \.*. In the previous examples, the command will apply to all the devices having IP address starting with 10.76.38.* [netmask] and 10.72.* [netmask].

As explained earlier, the regular expressions must be enclosed with [# #].

Example: + snmp-server location [#.*#]

The command will pass the compliance check for the following:

```
snmp-server location any_single_word_here
```

The command will fail the compliance check for the following:

```
snmp-server location more than one word
```

This happens because, with this template, the check will be performed only for one word after snmp-server location. To match any number of words after snmp-server location, the command needs to be modified slightly as follows:

```
+ [#snmp-server location .*#] which will do a compliance check for
all forms of snmp-server commands like snmp-server location location1
location2 location3, ... and so on.
```

Limitations of Compliance Management with Baseline Templates

Following are some limitations of using baseline templates:

- In a parameterized file, default values for devices cannot be provided.
- Currently, the job can be configured to send an email notification only when the job succeeds. There is no way to send a notification in case of network noncompliance.
- Conditional deployment is not possible in direct deployment. For instance, configuring the RCP or TFTP server in all the network devices only if they are not already configured is not possible using direct deployment.

- In case of Compliance and Deploy, the job status shows the results of the compliance. To help ensure a successful deployment, you need to check the device manually.
- Compliance and Deploy can be run as a single job with the following limitations:
 - Negative matches or absence of commands in the configuration would work fine.
 - Positive matches will result in unexpected results in the case of wildcard entries, for instance, to check for the presence of a community string in devices like + snmp-server community [#red.*#] RO. Choosing this template for Compliance and Deploy would deploy the same command to all the devices that do not have an RO community string starting with "red". At the end of the job, the devices that didn't have an RO community string that started with "red" will now have an RO community string called "[#red.*#]", which is not the expected result.

References

http://www.cisco.com/en/US/docs/net_mgmt/ciscoworks_resource_manager_essentials/4.2/user/guide/baseline.html.



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

CCDE, CCSI, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco Nurse Connect, Cisco Stackpower, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks. Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks, and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0903R)