# Sun Update Connection - Enterprise 1.0 Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–6533–11
September, 2006

060912@15490

# Contents

# Preface

This book contains procedures and explanations for starting, configuring, and troubleshooting Sun Update Connection – Enterprise.

## Who Should Use This Book

This book is intended for users, developers, and system administrators.

## How This Book Is Organized

Chapter 1 includes procedures for predefining hosts and configuring a browser on the console. It also includes command-line interface (CLI) commands needed to start, stop, restart,and check status of Sun Update Connection – Enterprise applications.

Chapter 2 describes update management and support for Solaris™ Operating System (Solaris OS) update cluster.

Chapter 3 describes how to edit the rc files and how to start the console with overriding rc parameters.

Chapter 4 describes how to integrate Sun Update Connection – Enterprise with IBM and with the Sun BMI server.

Chapter 5 describes how to use Sun Update Connection – Enterprise with Shared Resources support.

Chapter 6 includes technical explanations and procedures for use of Sun Update Connection – Enterprise in various network configurations.

Chapter 7 describes how the technology, and how to configure Sun Update Connection – Enterprise for Linux.

Chapter 8 describes the backup and recovery procedures for the system dependency server (SDS).

## Related Books

In addition to this book, find more information in the following publications.

**TABLE P–1** Related Information

| Information | Description |
|---|---|
| *Sun Update Connection – Enterprise Release Notes* | Describes any bugs and known problems. This book includes late new features and platform support. |
| *Sun Update Connection - Enterprise 1.0 Quick Start Guide: Getting Started* | Provides information about planning for an installation, and installing Sun Update Connection – Enterprise software. This book also includes step-by-step procedures for uninstalling the Sun Update Connection – Enterprise agent, console, or CLI. |
| *Sun Update Connection – Enterprise Bare Metal Installation Guide* | Describes the steps needed to install the software for a bare metal server. |
| *Sun Update Connection – Enterprise 1.0 User's Guide* | Provides information to system administrators about how to use the Sun Update Connection – Enterprise software to manage software updates to systems that run the Solaris Operating System or Linux distributions. |

## Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

---

**Note –** Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

---

## Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Sun will provide a technical FAQ in a read only format at `http://forum.sun.com/jive/forum.jspa?forumID=334`. The URL is subject to change and might be removed at Sun's discretion.

## Typographic Conventions

The following table describes the typographic conventions that are used in this book.

**TABLE P–2** Typographic Conventions

| Typeface | Meaning | Example |
|---|---|---|
| `AaBbCc123` | The names of commands, files, and directories, and onscreen computer output | Edit your `.login` file.<br>Use `ls -a` to list all files.<br>`machine_name% you have mail.` |
| **`AaBbCc123`** | What you type, contrasted with onscreen computer output | `machine_name%` **`su`**<br>`Password:` |
| *aabbcc123* | Placeholder: replace with a real name or value | The command to remove a file is `rm` *filename*. |
| *AaBbCc123* | Book titles, new terms, and terms to be emphasized | Read Chapter 6 in the *User's Guide*.<br>A *cache* is a copy that is stored locally.<br>Do *not* save the file.<br>**Note:** Some emphasized items appear bold online. |

## Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

**TABLE P–3** Shell Prompts

| Shell | Prompt |
|---|---|
| C shell | `machine_name%` |
| C shell for superuser | `machine_name#` |
| Bourne shell and Korn shell | `$` |
| Bourne shell and Korn shell for superuser | `#` |

CHAPTER 1

# Application Administration

This chapter describes how to predefine hosts in the console before you install the agent and how to configure a browser on the console. It also provides the commands used to start, stop or check the status of applications after installation.

This chapter covers the following topics:

- "Predefining Hosts" on page 11
- "Configuring Your Browser on the Console" on page 13
- "Executing CLI Commands" on page 13
- "Troubleshooting" on page 16

## Predefining Hosts

You can define managed hosts in the console before the systems are set up and the agent is installed.

- "To Predefine Hosts in the Console" on page 11
- "To Predefine Hosts With the CLI" on page 12

### ▼ To Predefine Hosts in the Console

**Before You Begin** Ensure that the system dependency server (SDS) and the console or CLI are installed. For information about installing the SDS, see Chapter 3, "Installing the System Dependency Server and Knowledge Base," in *Sun Update Connection - Enterprise 1.0 Quick Start Guide: Getting Started*.

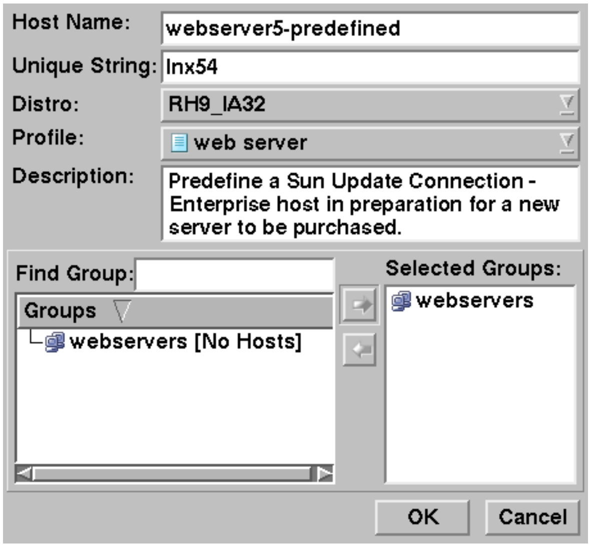
1 **Start the console with predefined host enabled.**

```
# uce_console -manual_host_create true
```

When the console opens, the Add Host button on the tool bar is enabled.

2 **Click Add Host.**

The Add Host window opens.



Host Name: webserver5-predefined

Unique String: lnx54

Distro: RH9_IA32

Profile: web server

Description: Predefine a Sun Update Connection - Enterprise host in preparation for a new server to be purchased.

Find Group:

Selected Groups: webservers

3 **In the Host Name field, type a name for the host.**

4 **In the Unique String field, type the unique string to be reserved for the host.**

5 **Select the distribution to be installed on the host.**

6 **(Optional) Select a profile to attach to the host.**

7 **(Optional) In the Description field, type the description for the host.**

8 **(Optional) In the Groups section, select a group to which you want the host to belong.**

9 **Click OK.**

The host name appears in the Hosts list, with the yellow Not Connected icon.

## ▼ To Predefine Hosts With the CLI

Using the CLI, the Sun Update Connection – Enterprise *hostname* is the Unique String. You can change the host name later, but the Unique String should be the real string of the host.

**Before You Begin** Ensure that the system dependency server (SDS) and the console or CLI are installed. For information about installing the SDS, see Chapter 3, "Installing the System Dependency Server and Knowledge Base," in *Sun Update Connection - Enterprise 1.0 Quick Start Guide: Getting Started*.

1 **Open a terminal window.**

2 **Type the following command:**

```
# uce_cli -ah -D distribution -h hostname -u username -p password
```

# Configuring Your Browser on the Console

The console can open a web browser to specific links. For example, from the Component Information window of a component there might be a link to its web advisory. Click the Browser button to see the advisory. Before you can do this, you need to configure the console to find your browser.

## ▼ To Configure Your Browser on the Console

1 **Start the console.**

2 **Select Preferences from the Tools menu.**

The Preferences window opens, with the Console-Files preferences shown.

3 **In Browser text box, type the name of the browser you want to use.**

4 **In the Path text box, type the path to the browser.**

If you do not know the path, in a command-line terminal window, type the `which` *browser* command.

For example, `which firefox` might return `/usr/local/firefox-installer/firefox`.

# Executing CLI Commands

The command-line interface commands start with **`uce_cli`**. To get help, type the `man uce_cli` command.

To execute the command in the command line, type **`uce_cli -`**<command> [**-**<parameter> <value>] [...] [-<flag>] [...]

To execute the command in a script, type the commands in a shell script, just as you would embed other Linux commands.

# Summary of Commands

This section lists command-line operations used to start, stop, restart, or check the status of any of the applications after installation. They all start up automatically after installation, except for the console. Status checks and restart commands are helpful when troubleshooting an application.

The following sections contain the command syntax for the different applications:

- "Starting Applications" on page 14
- "Stopping Applications" on page 15
- "Restarting Applications" on page 15
- "Checking the Application Status" on page 16

## Starting Applications

The agent starts automatically, but if needed, you can restart the agent.

The following table provides the syntax for the commands that start the different applications.

**TABLE 1–1** Starting Applications

| Application | Syntax |
|---|---|
| To start the console | `uce_console` |
| To start the agent | `/etc/init.d/uce_agent start` |
| To start dependency manager | `/etc/init.d/uce_engine start` |
| To run the CLI | `uce_cli`<br>(without a parameter, a list of commands is displayed) |
| To start the SDS | `/etc/init.d/uce_server start` |

**Note –** If you receive a failed login request error message when attempting to log into the console, see "Troubleshooting" on page 16.

**EXAMPLE 1–1** To Start a Console

The console is the interface to the other applications, the control center for Sun Update Connection – Enterprise.

1. On the console system, type `uce_console`.
2. Wait until the Login window appears.

   If this is the first time you have run Sun Update Connection – Enterprise, and there are no local users yet, log in as `admin`, with the password `123`. After the first login, you are prompted to change the password.

EXAMPLE 1–1 To Start a Console *(Continued)*

When the console appears, you can begin managing hosts.

## Stopping Applications

The following table provides the syntax for the commands that stop the different applications.

TABLE 1–2 Stopping Applications

| Application | Syntax |
|---|---|
| To stop the agent | `/etc/init.d/uce_agent stop` |
| To stop the dependency manager | `/etc/init.d/uce_engine stop` |
| To stop the SDS | `/etc/init.d/uce_server stop` |

EXAMPLE 1–2 To Stop an Agent

1. Open a terminal window on the managed host.
2. Type the following command:

```
# /etc/init.d/uce_agent stop
```

## Restarting Applications

The following table provides the syntax for the commands that restart the different applications.

TABLE 1–3 Restarting Applications

| Application | Syntax |
|---|---|
| To restart the agent | `/etc/init.d/uce_agent restart` |
| To restart the dependency manager | `/etc/init.d/uce_engine restart` |
| To restart the SDS | `/etc/init.d/uce_server restart` |

EXAMPLE 1–3 To Restart an Agent

1. Open a terminal window on the managed host.
2. Type the following command:

```
# /etc/init.d/uce_agent restart
```

### Checking the Application Status

The following table provides the syntax for the commands that check the application status.

**TABLE 1–4** Checking the Application Status

| Application | Syntax |
|---|---|
| To get the status of an agent | `/etc/init.d/uce_agent status` |
| To get the status of the dependency manager | `/etc/init.d/uce_engine status` |
| To get the status of the SDS | `/etc/init.d/uce_server status` |

**EXAMPLE 1–4** To Check the Status of an Agent

1. Open a terminal window on the managed host.
2. Type the following command:

```
# /etc/init.d/uce_agent status
```

# Troubleshooting

This section explains how to correct a failed login request.

## Failed to Send Login Request

When you try to log into the console, the login fails and the following message is displayed: `Failed to start. Reason: Cannot send login request.`

This message is displayed if the dependency manager (DM) is down.

### ▼ To Correct a Failed Login Request

1 **Type the following to restart the DM.**

```
# /etc/init.d/uce_engine restart
```

2 **Login to the console.**

◆ ◆ ◆ CHAPTER 2

# Managing Solaris Software Updates

Sun Update Connection – Enterprise offers new features for Solaris update management. This chapter covers the following topics:

- "Overriding Sun Recommendations" on page 17
- "Supporting Solaris Update Clusters" on page 22

## Overriding Sun Recommendations

The default behavior of Sun Update Connection – Enterprise is to deploy Solaris updates according to Sun recommendations. Deploying updates through Sun Update Connection – Enterprise could necessitate numerous restarts, restart and reconfigure, or manual deployments during the process of the job.

You have the option to override the recommendations to restart, restart and reconfigure, or deploy manually, during a job. You can set Sun Update Connection – Enterprise to ignore these recommendations and force the job to finish deployments before you restart (or otherwise manually apply the update management).

**Caution** – It is **not** recommended that you override the default update deploy and apply actions set up by Sun. If you follow this procedure, realize that you take responsibility for the results.

### Overriding Sun Update Policy Recommendations

You override specific recommendations for each Solaris host in the Preferences window, in the Update Policy category. If your Host Preferences do not include Update Policy on the left side, contact technical support.

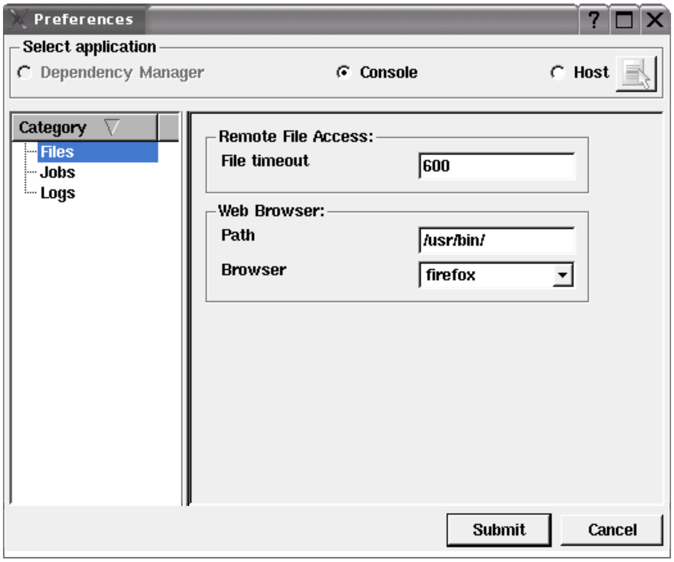
**Note –** When you select a host in the Preferences window, you have the option to select from any of the managed hosts in the system. However, this procedure is relevant only for Solaris hosts.

## ▼ To Override Sun Update Policy Recommendations

1 **Open the Preferences window.**

You can open the preferences window using one of the following options:

- From the tool bar, click the Preferences button.
- From the Tools menu, choose Preferences.



Preferences

Select application: ○ Dependency Manager ● Console ○ Host

Remote File Access:

File timeout: 600

Web Browser:

Path: /usr/bin/

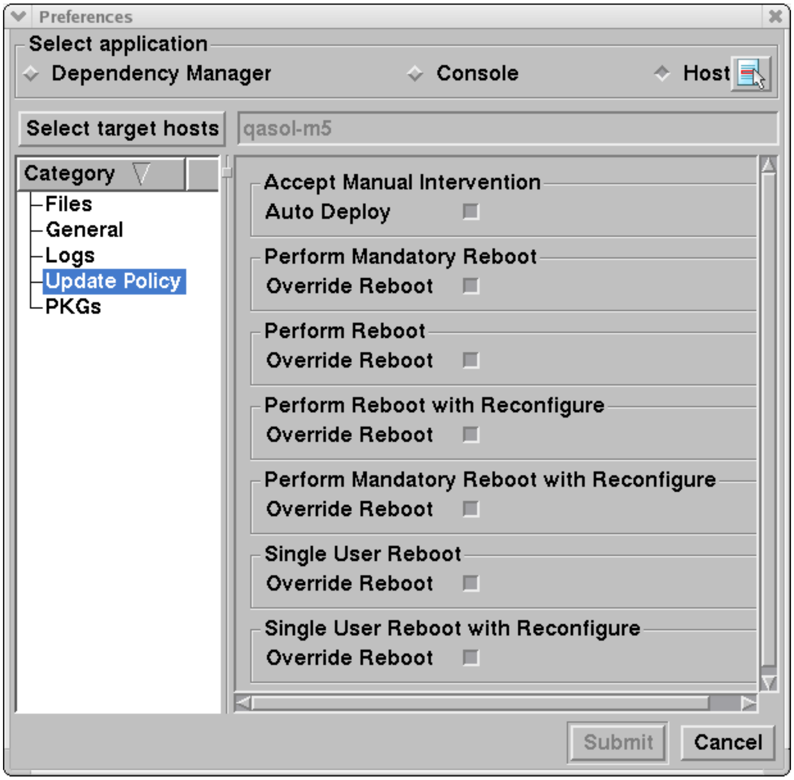
Browser: firefox

2 **Select a single Solaris host, then click OK.**

Wait while the Preference values of the selected agent are uploaded to the console.

**Note –** Preference values can be uploaded only if the agent is not busy with another command.

3 **In the Preferences window left-panel, click Update Policy.**



qasol-m5

Accept Manual Intervention
Auto Deploy □

Perform Mandatory Reboot
Override Reboot □

Perform Reboot
Override Reboot □

Perform Reboot with Reconfigure
Override Reboot □

Perform Mandatory Reboot with Reconfigure
Override Reboot □

Single User Reboot
Override Reboot □

Single User Reboot with Reconfigure
Override Reboot □

4 **If you want to set the override preferences for multiple hosts, click Select target hosts and then select the hosts.**

5 **Select the check boxes of the Sun recommendations that you want to override:**

- **To deploy an update that has a recommendation for manual intervention, select Auto Deploy to deploy updates automatically in the job.**
- **To deploy one or more updates that recommend that the host be restarted after the update is deployed, select the Override Restart option of the specific notification. By selecting the Override Restart option, you can complete all deployments before restarting the host. To apply the updates, restart the host after all updates are deployed.**

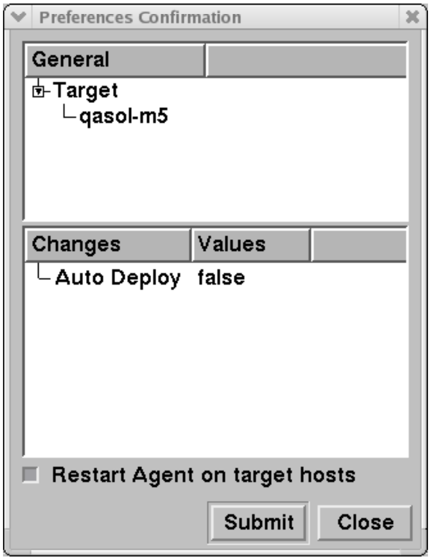
---

**Note –** If an update requires reconfiguration or single-user deployment, perform the necessary management tasks after the job has been completed.

---

6 **Click Submit.**

The Preferences Confirmation window opens.



Target
qasol-m5

| Changes | Values |
|---|---|
| Auto Deploy | false |

☐ Restart Agent on target hosts

7 **Check that the changes are appropriate, select the Restart Agent check box, then click Submit.**

## Handling Jobs With Override Settings

After you turn on the overrides, the next update management job that you run in Sun Update Connection – Enterprise on the affected hosts will use the override settings. Note that the overrides do not affect the Restart or Restart and Reconfigure profiles; they affect only the Sun update management recommendations.

While the job is running, you can monitor the progress of the update deployments. Notice that the restart and manual intervention actions are listed in the Host Progress window, and that they are marked as successfully done, although your overrides ensure that these actions are not actually done.

### ▼ To Monitor an Update Management Job

1 **To view the Jobs list, select View, then select Jobs.**

2 **From the Jobs list, select an update management job.**

The tasks of the job appear in the Tasks list.

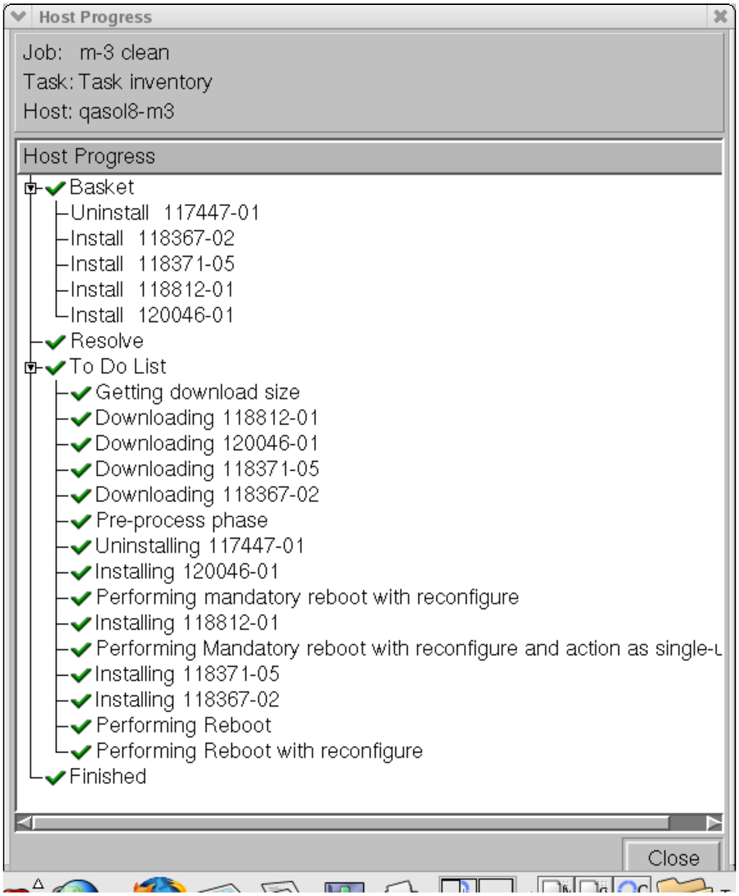
3 **From the Tasks list, select a task.**

The hosts running this task appear in the Hosts list of the Jobs panel.

4 **Select a Solaris host.**

5 **From the Jobs menu, choose Hosts, then choose Show Progress.**

The Host Progress window opens.



Job: m-3 clean
Task: Task inventory
Host: qasol8-m3

Host Progress

- ✓ Basket
  - Uninstall 117447-01
  - Install 118367-02
  - Install 118371-05
  - Install 118812-01
  - Install 120046-01
- ✓ Resolve
- ✓ To Do List
  - ✓ Getting download size
  - ✓ Downloading 118812-01
  - ✓ Downloading 120046-01
  - ✓ Downloading 118371-05
  - ✓ Downloading 118367-02
  - ✓ Pre-process phase
  - ✓ Uninstalling 117447-01
  - ✓ Installing 120046-01
  - ✓ Performing mandatory reboot with reconfigure
  - ✓ Installing 118812-01
  - ✓ Performing Mandatory reboot with reconfigure and action as single-u
  - ✓ Installing 118371-05
  - ✓ Installing 118367-02
  - ✓ Performing Reboot
  - ✓ Performing Reboot with reconfigure
- ✓ Finished

6 **The To Do list might include** `Performing <action>` **items, such as** `Performing Restart, or Performing Restart and Reconfigure`**. These items are marked with a green check, but they are not actually done.**

You might see `Attention! <action>`, with an action that you would normally have to perform before the job continues. With the relevant override turned on, these items are also marked with a green check and the job continues.

At the end of the job, be sure that you complete the actions that are required for each update.

### To View the Job Log

From the Jobs menu, choose Hosts, then choose Show Log.

The Job Log window opens.

The Job Log shows the status of both complete and incomplete deployment updates. It shows the actions that should have been done, according to the Sun recommendations, as well as actions that were not done. For example, if an action was not done, you might see the following message: `Overridden by User Request`.

When the update management is done, be sure to do the appropriate actions.

**Caution –** Remember that the overrides are set by agent, not by job. If you want another job to run according to Sun recommendations (restarting and otherwise managing the host in the midst of a job), deselect the Update Policy preferences.

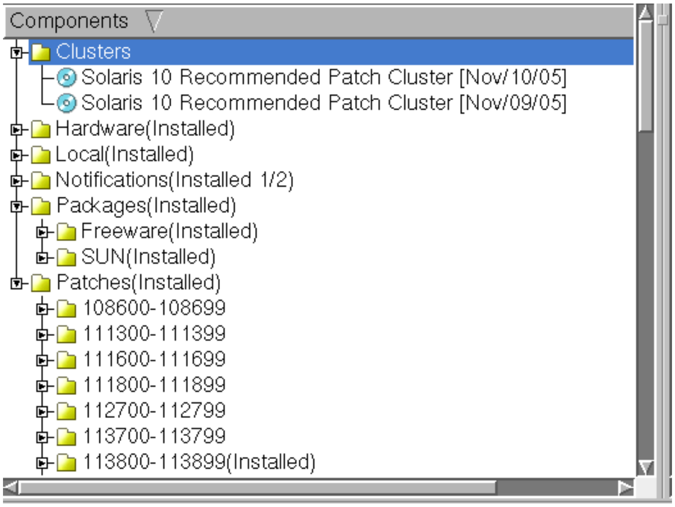
## Supporting Solaris Update Clusters

Sun publishes Update Clusters, which are bundles of updates, provided for a single installation. Without Sun Update Connection – Enterprise, most UNIX administrators plan for downtime to install Update Clusters, because some of the updates require single-user mode. Some administrators research the contents of a cluster, and install only those updates that are not already installed.

With Sun Update Connection – Enterprise, you can install Update Clusters while the hosts are still up, and you can let Sun Update Connection – Enterprise do the research and decide what needs to be installed.

The previous section explained how to set overrides to Sun recommendations for restarting, which includes single-user installations and reconfigurations. In general, do not use the overrides unless you are prepared to take responsibility for the results. However, when deploying Update Clusters, the most efficient method is to turn on all of the overrides.

Sun releases Update Clusters, as required, to add updates to the cluster. Theconsole includes a Clusters category, which contains the latest Update Clusters.



```
Components
Clusters
  Solaris 10 Recommended Patch Cluster [Nov/10/05]
  Solaris 10 Recommended Patch Cluster [Nov/09/05]
Hardware(Installed)
Local(Installed)
Notifications(Installed 1/2)
Packages(Installed)
  Freeware(Installed)
  SUN(Installed)
Patches(Installed)
  108600-108699
  111300-111399
  111600-111699
  111800-111899
  112700-112799
  113700-113799
  113800-113899(Installed)
```

You can create a profile which deploys the relevant Update Clusters and checks them for new updates on a schedule. See the *Sun Update Connection – Enterprise 1.0 User's Guide* for instructions on creating profiles and on scheduling profiles.

If you schedule a profile for the installation of an Update Cluster, the deployment is not actually done unless it is needed. Sun Update Connection – Enterprise determines if the selected hosts already have the most recent cluster updates. If so, the profile job ends successfully, with nothing done. If one or more updates are added to the cluster, Sun Update Connection – Enterprise installs them on your hosts. All of this takes place in the background with no down-time.

## Withdrawn Updates in Clusters

When Sun publishes an Update Cluster on `http://www.sun.com`, only the latest version is published. This is the recommended cluster to install.

If a job to install an updatecluster is marked as failed, the job might include a withdrawn update. To troubleshoot the job, determine if the cluster includes updates that Sun withdrew. Withdrawn updates cannot be maintained by Sun Update Connection – Enterprise, because they are not maintained by Sun. If a cluster includes a withdrawn update, the cluster cannot be installed.

In some cases, an update is withdrawn just minutes after a cluster is published. For this reason, check the cluster for withdrawn updates before attempting to install the cluster.

If the latest cluster published by Sun has a withdrawn update, wait for the next publication.

## ▼ To Check a Cluster for Withdrawn Updates

**Before You Begin** Ensure that the Inventory panel of the main window is available by selecting View, then selecting Inventory.

1. **From the Components list, right-click a cluster and choose Details.**

   The Details window opens.

2. **Open the Rules tab and expand the list of updates of the cluster.**

   If even one of the updates is marked with `WITHDRAWN`, the cluster cannot be installed.

CHAPTER 3

# Customizing Configuration Files

This chapter explains the configuration settings for customizing the operations of Sun Update Connection – Enterprise through the command-line. In addition, this chapter explains how to configure the Apache logs of the server application, outside of the configuration files.

This chapter covers the following topics:

- "Viewing and Changing Configuration Files" on page 25
- "Overriding Configurations" on page 26
- "Configuring Sun Update Connection – Enterprise to Create Apache Logs" on page 27

## Viewing and Changing Configuration Files

This section explains how to access the `uce.rc` configuration file. It is strongly recommended that you do not change this file; open for viewing only. To make changes to the configuration of an application, change the `.uce.rc` override file.

The server, Dependency Manager (DM), agents, and the console each has a `uce.rc` file:

- Server - `/usr/local/uce/_server/cgi_bin/uce.rc`
- Dependency Manager - `/usr/local/uce/engine/bin/uce.rc`
- Agents (on the host) - `/opt/local/uce/agent/bin/uce.rc`
- Console - `~/.uce_console/uce.rc` (where ~ is the home of the installation user)

  The most common configuration parameters are offered for viewing and changing from within the console, in the Settings window.

  You can overwrite other values of the factory default configuration file with the `.uce.rc` override file, which is stored in the same directory as the `uce.rc` file.
- `uce.rc` file - Complete list of factory default parameters and values
- `.uce.rc` file - User configured environment created by `ezInstaller`, contains only those parameters that override the factory defaults

To view all parameters, open the `uce.rc` file. To override a parameter, copy the parameter line from the default file (close without saving) to the `.uce.rc` file and make the change in the parameter value.

The following table shows the commands to open override files.

**Note –** Use a text editor, such as emacs, gedit, or vi to edit the files. The vi text editor is used in this document.

| Configuration File | Syntax |
|---|---|
| to open console configuration file | `vi ~/.uce.console/.uce.rc`<br>If you log into the console as a user, this directory is your **/home**.<br>If you log into the console with root permissions, this directory is under **/root**. |
| to open agent configuration file | `vi /opt/local/uce/agent/bin/.uce.rc` |
| to open dependency manager (DM) configuration file | `vi /usr/local/uce/engine/bin/.uce.rc` |
| to open Server configuration file | `vi /usr/local/uce/server/cgi_bin/.uce.rc` |

When you change a value with a copied parameter to `.uce.rc`, the new value is always applied, until you delete the line. The next section explains how to override configuration values for one session.

# Overriding Configurations

You can start the console with override values, which supersede `.uce.rc` values (and therefore also the `uce.rc` values).

To find the correct parameter name, open the `uce.rc` file (but do not change this file!). If you look at a line from the right (the end) and go leftward, the parameter name ends to the left of the comma (separating the parameter and the value), and begins to the right of the first dot that you encounter:

```
( invisible.debug.__group1.debug_mode, false );
```

You can start the console from the command line, with an explicit parameter value, for the current session. The parameter value reverts to the `.uce.rc` value (or to the `uce.rc` value if it does not appear in the override file) when you restart the application.

Syntax: **`uce_console -<parameter> <value>`**

**`uce_console -update_director_mode true`**- For updating remote agents

**`uce_console -debug_mode true -debug_log_level 5`**- For debugging; technical support might ask you to turn on these parameters, to help them find and fix problems.

# Configuring Sun Update Connection – Enterprise to Create Apache Logs

The server application in the system dependency server can create logs for the embedded Apache server. To regulate the size of the Apache-Sun Update Connection – Enterprise logs, go to the logrotate file; controls over these specific logs are not in the `uce.rc` file.

## ▼ To Control Apache Sun Update Connection – Enterprise Logs

1. **Check if you have the logrotate file: /etc/logrotate.d/uce_server**

   If this file is not on your system, create a new file with this path name.

   If you do not have the logrotate application, use Sun Update Connection – Enterprise to install it on the SDS system. Use a cron job to run the log rotations hourly, or daily.

2. **Open `/etc/logrotate.d/uce_server` for editing.**

   Its content should include:

   ```
   /var/log/uce_server/access_log
   /var/log/uce_server/error_log
   /var/log/uce_server/ssl_engine_log
   /var/log/uce_server/ssl_request_log
   {
       rotate 2
       missingok
       size=100M
       compress
       sharedscripts
       postrotate
           /etc/init.d/uce_server graceful
       endscript
   }
   ```

   If this content is not in the file, add it.

3. **To have more rotating files, increase the value of the `rotate` parameter.**

4. **To ensure your logs are small, decrease the value of the `size` parameter.**

CHAPTER 4

# Integrating With Other Tools

This chapter describes how Sun Update Connection – Enterprise can be integrated with other tools. Integration provides Sun Update Connection – Enterprise features with an IBM Tivoli software solution, bare metal Linux installation, and an on-demanding computing implementation.

This chapter covers the following topics:

- "Integrating with IBM" on page 29
- "Integrating With a Bare Metal Installation Server" on page 30

## Integrating with IBM

Sun Update Connection – Enterprise has been approved as ServerProven for IBM eServer.

In addition, Sun Update Connection – Enterprise is validated as Ready for IBM Tivoli Intelligent ThinDynamic Orchestrator.

The Sun Update Connection – Enterprise device driver for IBM Tivoli package comes with user documentation, `~/doc/readme.html`, and three workflows:

- `OnStage_Get_Job_Status.wkf`- Outputs current status of a given job
- `OnStage_Security_Check_Job.wkf` – Executes a check on all installed software components of a given system, installing available security updates or upgrading to secured versions as needed
- `OnStage_Submit_Profiled_job.wkf` - Sends a job that deploys an Sun Update Connection – Enterprise profile (a list of required and non-allowed components) on a given system or group of systems

  The Sun Update Connection – Enterprise driver implements the software.Install logical operation of the IBM Tivoli Intelligent ThinDynamic Orchestrator.

  These workflows execute various Sun Update Connection – Enterprise functions from within the Tivoli Intelligent ThinDynamic Orchestrator. Each workflow sends a job request or a get-status request to the Sun Update Connection – Enterprise SDS, with internal, secured connections between the Tivoli initiator, the SDS, and the relevant managed host.

**Note –** The Ready for IBM Tivoli Software Mark, Ready for IBM eServer Mark, and the IBM ServerProven Mark, and the trademarks contained therein are trademarks of IBM Corp. IBM is not the licensor of Sun Update Connection – Enterprise and does not make any warranties regarding this product.

# Integrating With a Bare Metal Installation Server

The Bare Metal Installation (BMI) Server automates Linux distribution and operating system installation on a bare-metal system. The BMI Server integrates automatically with Sun Update Connection – Enterprise to provide deployment features such as dependency handling, use of the Dependency Resolver to avoid conflicts, and system checks with the predefined profiles.

The BMI Server provides a list of Linux distributions to choose from. It installs the one that you selected on a managed system, whether it is bare or already provisioned.

In addition, you can choose a profile from the list in your local Sun Update Connection – Enterprise. The BMI Serverinstalls the Linux operating system and the selected profile at the same time. You can create a web server, or the profile that you want.

Contact Sun Customer Support for BMI information and documentation.

For information about installing a Bare Metal Server, go to *Sun Update Connection - Enterprise 1.0 Bare Metal Installation Guide*.

CHAPTER 5

# Shared Resources

This chapter explains how to use Sun Update Connection – Enterprise with Shared Resources support for z/VM platforms.

On a zSeries S/390 running Virtual Systems from VM/ESA V2R4, z/VM V3R1, or later, you can create minidisks. A minidisk is allocated space consisting of consecutive cylinders on a specified DASD volume, which acts as a separate device. A minidisk is owned by one VM user; it cannot exist without the virtual user. Only one minidisk owner has read-write access at any time; other users may share it with read-only access.

Sun Update Connection – Enterprise supports sharing only of the `/usr` and `/opt` directories of the minidisks in their Linux environment. The Linux image running on a minidisk must have version 4 of RPM.

This chapter covers the following topics:

- "Defining Master Hosts and Sharing Hosts" on page 31
- "Sharing Resources - Why and How" on page 32
- "Using Master Groups and Custom Groups" on page 36
- "Deploying Jobs With Sharing Hosts" on page 37

## Defining Master Hosts and Sharing Hosts

A VM user is a definition of resources, some real and some virtual, allocated for use or sharing by one username.

```
USER LNXDLB01 CORP 64M 128M G
```

| | |
|---|---|
| 00142 | INCLUDE CMSPROF |
| 00143 | IUCV ALLOW |

| | |
|---|---|
| 00144 | IUCV ANY |
| 00145 | OPTION QUICKDSP |
| 00146 | ACCOUNT XXXXXXX CORP |
| 00147 | NOPDATA |
| 00148 | DEDICATE 5C7 5C3 |
| 00149 | DEDICATE 5C8 5C4 |
| 00150 | LINK MAINT 291 291 RR |
| 00151 | MDISK 0191 3390 1 1 WORK04 MR |
| 00152 | MDISK 0208 3390 001 1100 WORK0B MR |
| 00153 | MDISK 0209 3390 001 1100 WORK0C MR |
| 00154 | MDISK 0201 3390 001 1100 WORK05 MR |
| 00155 | MDISK 0203 3390 451 150 WORK02 MR |

A *master host* is a VM user, running a Linux image that includes an Sun Update Connection – Enterprise agent, and that can share a minidisk with other Linux images.

A *sharing host* is a VM user, running a Linux image that includes an Sun Update Connection – Enterprise agent, that has read-only access to the minidisk that a master host shares with it.

Master hosts have full access to the resources of the Linux image of the VM user. sharing hosts have links to the resources of a master host.

## Sharing Resources - Why and How

Sharing resources saves space by sharing duplicate files across a Linux network. For example, you might have 30 networked systems all running Linux. The systems are grouped by function, so some of them have directories that are almost identical to the other computers in their group. If you think of these systems as VM users on a zSeries running Linux, you can save space by not sharing directories.

To save space, and to make like systems consistent, specific directories on minidisks can be shared. Sun Update Connection – Enterprise automatically locks these directories, so that the owner of the minidisk has full access, but sharing hosts have read-only access. Automatically fulfilling this requirement saves the VM operator time and makes sure the system is secure and consistent.

Another Shared Resources requirement is that the sharing hosts have specific mount points to the shared minidisk. This is also fulfilled automatically by Sun Update Connection – Enterprise.

## Mount Points for Shared Directories

Sun Update Connection – Enterprise manages software, so efficient sharing in an Sun Update Connection – Enterprise system concentrates on software-relevant directories. Sun Update Connection – Enterprise shared resources support is relevant to Linux operating system directories, not user-defined directories.

The most important shared directories are `/usr` and `/opt`.

The `/usr` directory contains system libraries, administrative commands and files, and the main files for installed applications. When you install an application, its main parts go in the `/usr` directory. Other directories hold the configuration files and data, the type of things that you want to sometimes change. Sharing the `/usr` directory means that multiple VM users (master host and sharing hosts) can use the same applications, although the largest part of the package is be installed only once (on the master host).

The directories that you want to share must be mounted in `/proc/mounts` of both the master host and the sharing hosts.

## Master Host and Sharing Host Synchronization

When you install an application on a Linux image, some files are installed in directories other than `/usr`. Therefore, when you use Sun Update Connection – Enterprise to install an application on a master host, the same job must run on all of the sharing hosts of the master host. The application is installed only once, but every sharing host requires the dependent components installed in other directories.

Failing to maintain duplicate installation procedures between a master host and its sharing hosts causes synchronization errors. Such errors disable management on the unsynchronized sharing hosts. Sharing hosts that do not perform the same installation job as its master host cannot function as a managed host of Sun Update Connection – Enterprise.

To automatically maintain synchronization, Sun Update Connection – Enterpriselimits jobs that include sharing hosts..

The following synchronization limits apply:

- You cannot send a job to only one or some of the sharing hosts of one master host; you must send it to the master host only. Sun Update Connection – Enterprise ensures that the sharing hosts also receive the job.
- RPM jobs must be sent to a master host that is online. If the master host is down, the job fails.
- Every managed host must have either zero or one master host. It must either be a master host or have only one master.

## ▼ To Create a Master Host

1. **Define a VM user with read or write access to some minidisks.**
2. **At the end of the profile exec a file, type the line: `ipl <dev #>`**
3. **Upload Linux (with the DDR utility) onto <dev #>.**
4. **Start the new VM user from any 3270 emulation.**
5. **Change the TCP/IP of this new Linux image with the `ifconfig` command.**
6. **Fix the routing table and then connect with telnet to the new VM user.**
7. **Change the network configuration with YAST (5-6) to be permanent.**
8. **Restart Linux.**
9. **Copy the agent.tgz file to the new VM user Linux operating system.**
10. **Untar the file and type `./Install` to start the agent `ezInstaller`.**
11. **After the agent is installed, view the agent configuration file.**

    ```
    vi /opt/local/uce/agent/bin/uce.rc
    ```

12. **Find the `mf_shared_path` parameter and check the value for this parameter. The value should be `/usr` or `/opt`, according to the directory that you want this sharing host to access from its master.**

    If the value is incorrect, copy the entire line of this parameter to the **.uce.rc** file and edit the value to be the correct directory. When the Sun Update Connection – Enterprise agent is installed on this VM user, you have a master host.

## ▼ To Create a Sharing Host

1. **Define a VM user that, in addition to its own minidisks, has links to the minidisks of another VM user, with read-only access.**

    The VM user that controls the linked resources is the master host of this sharing host.

2. **At the end of the profile exec a file, type the line: `ipl <dev #>`**
3. **Upload Linux (with the DDR utility) onto <dev #>.**
4. **Start the new VM user from any 3270 emulation.**

5 **Edit the /boot/parm file: add `(ro)` near dev # items that are owned by the master host. Example:** `dasd=xxxx-yyyy, 202 (ro), zzz`

6 **Edit the /etc/fstab file: specify read-only mount points for the shared disks; make sure that the last two parameters on the line are “0 0”.**

7 **Do a silo/zilo.**

8 **Change the TCP/IP of this new Linux image with the `ifconfig` command.**

9 **Fix the routing table and then connect with telnet to the new VM user.**

10 **Change the network configuration with YAST (9-10) to be permanent.**

11 **Restart Linux.**

12 **Copy the `agent.tgz` file to the new VM user Linux operating system.**

13 **Untar the file and type `./Install` to start the agent `ezInstaller`.**

14 **After the agent is installed, view the agent configuration file.**

```
vi /opt/local/uce/agent/bin/uce.rc
```

15 **Find the `mf_shared_path` parameter and check the value for this parameter. The value should be `/usr` or `/opt`, according to the directory that you want this sharing host to access from its master.**

If the value is incorrect, copy the entire line of this parameter to the `.director.rc` file and edit the value to be the correct directory. When the installed agent starts, it discovers that it is running on a shared disk environment and registers its VM user as a sharing host on the system dependency server. When an agent starts, a master host writes its master signature in **/<shared_path>/.aduva_master**

A sharing host attempts to write to the same path. It finds that it does not have write permissions, reads the .aduva_master file and discovers both that it is a sharing host and who is its master.

## ▼ To Unregister a Sharing Host From the System Dependency Server

1 **Take the sharing host offline.**

2 **In the Hosts window, select the host from the All Hosts Group.**

3 **Click Delete.**

4 **In the confirmation dialog box that opens, click Delete.**

### To Unregister a Master Host From the System Dependency Server

**Before You Begin** Ensure that the master host and all of its sharing hosts are offline.

1. **In the All Hosts Group, select the Master Host, then click Delete.**

2. **In the confirmation dialog box that opens, click Delete.**

3. **Select the icon for the empty Master Group, then click Delete.**

## Using Master Groups and Custom Groups

A group is a set of managed hosts to which you can send jobs simultaneously. There are two types of groups:

- **Master Group**. Contains the master host and all its sharing hosts. A job cannot be sent to only one of the hosts in a Master Group; the entire group must receive any job.
- **Custom Group**. Contains any set of master hosts that you choose. If the hosts do not have sharing hosts, the group can run and successfully deploy jobs. If any one of the hosts in a Custom Group has sharing hosts, jobs cannot be deployed on this group.

When you create privileged users with Permitted Groups, the groups must be complete Master Groups. You cannot assign a Custom Group as a permitted group, even it contains only hosts without sharing hosts.

The Hosts window displays icons to indicate the type of the groups and the type of managed host.

**TABLE 5–1** Hosts Window and Groups Tab Icons

| Icon | Name | Description |
|---|---|---|
| | Default Group | All Hosts group, or distribution groups. |
| | Custom Group | User-defined logical group of managed hosts. Use Custom Groups to organize the system for easy management. Custom Groups can contain the following:<br>■ A host list of any master hosts you choose<br>■ Nested Custom Groups |
| | Master Group | Automatically created group of master host and the sharing hosts that have read-only access to the master's resources. To maintain synchronization, use the Master Group. The Master Group must meet the following requirements:<br>■ Contain only the master and sharing hosts<br>■ Cannot contain nested Custom Groups |

**TABLE 5–1** Hosts Window and Groups Tab Icons *(Continued)*

| Icon | Name | Description |
|---|---|---|
| | Master Host | An individual Linux operating system with full access to all its directories, with an installed agent, and maintaining an operational connection to the dependency manager. |
| | Unsynchronized Master Host | Master host has at least one sharing host that is not synchronized<br>Click Synchronize to fix this problem. |
| | Disconnected Master Host | Master host that has lost its connection to the DM. |
| | sharing host | A minidisk with read-only access to the directories of a master host that is configured to share its resources with this minidisk. |
| | Disconnected Sharing Host | A managed host that is not currently connected to the dependency manager. |

**Note –** Sun Update Connection – Enterprise creates a Master Group only after a master host has at least one sharing host. Every Master Group contains at least two managed hosts: the master host and the sharing host.

The Hosts window also shows a new button: **Synchronize**. Use this feature to create a job that runs only on sharing host, to synchronize it with its master host. This button is available only while Sun Update Connection – Enterprise is in Sharing Mode; it is enabled if you select a sharing host that is not synchronized with its master.

In the Hosts window, when you have selected either a sharing host or a Master Group, the New and Properties buttons are disabled.

# Deploying Jobs With Sharing Hosts

In an Sun Update Connection – Enterprise system without sharing hosts, you can run any job on any host or any group. You can create Custom groups of any combination of master hosts.

Once you have created a sharing host, there are limitations to the jobs that you can deploy.

## Job Restrictions

Synchronization between the sharing hosts and their master host must be maintained. Therefore, jobs that include sharing hosts have special limitations, to ensure and to troubleshoot synchronization.

The following limitations apply to deploy jobs with a host list that includes at least one sharing host:

- The complete Master Group must be included on the host list
- Only one job can be run on a Master Group at any time
- After the Master completes a job, *all* of its sharing hosts must complete the job

# Job Processes

After Sun Update Connection – Enterprise confirms that the entire Master Group has been selected for a job with sharing hosts, the job begins.

1. The job is sent to the master host on the host list.
2. The agent of the Master runs the Dependency Resolver and job procedures.
3. On success:
   - If the Masters in the host list do not have sharing hosts, the job ends.
   - If it is a Simulate mode job, Sun Update Connection – Enterprise checks the sharing hosts and then the job ends.
   - If it is a Deploy job, Sun Update Connection – Enterprise checks that the sharing hosts are synchronized with their master host. If this check is successful, the sharing hosts take the job solution that the master host used and run the same tasks. sharing hosts do not try to find a new solution; they must follow the same steps that their Master took.

## If a Problem Occurs

| Unsynchronized Sharing Host | |
|---|---|
| Situation | An error is given for a sharing host in the Status window, stating that the host is not synchronized with its Master. |
| Explanation | A sharing host was unable to successfully complete a job and is no longer synchronized with its master host. |
| Solution | 1. Select the sharing host in the Hosts window. The **Synchronize** button is enabled.<br>2. Click **Synchronize**.<br>A job beings that works on only this sharing host. It makes any changes necessary to make the software configuration of the sharing host compatible with the master host. The Synchronize Job feature is available only to an user with full permissions, and can be run on only one sharing host at a time. |

◆ ◆ ◆ CHAPTER 6

# Configuring Web Proxies and Firewalls

This chapter explains how Sun Update Connection – Enterprise works through web proxies and firewalls, and what you need to do to configure Sun Update Connection – Enterprise to work in such an environment.

The Sun Update Connection – Enterprise agent, console, CLI, and API communicatesimilarly. Generally, tThe terms console, CLI, or API can replace agent when discussing proxies and firewalls..

Between the Sun Update Connection – Enterprise components, communications are secured. Between the agents and the dependency manager (DM), communications are encrypted with an RSA private/public key pair. Messages between the Sun Update Connection – Enterprise Apache-based server and the agents are in the secure HTTPS protocol.

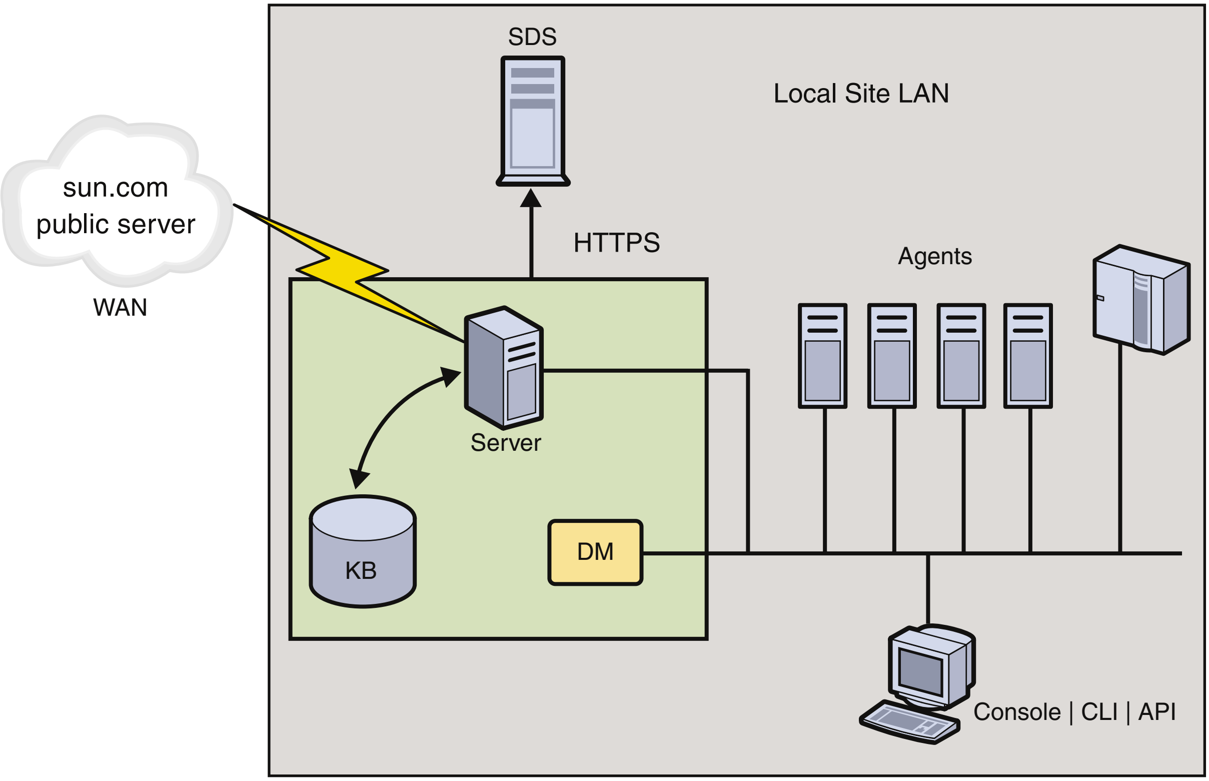
Local security measures might include:

- Firewall and web proxy between the Sun Update Connection – Enterprise server and the Sun server over the Internet.
- Routers between agents on one side, and the DM and Sun Update Connection – Enterprise server on an other, over different local networks.

This chapter covers the following topics::

# Defining the Components

The Sun Update Connection – Enterprise architecture includes the server, DM, agents, consoles, CLI, and API on the local side; and the Sun Update Connection – Enterprise public servers over the Internet. This section describes the architecture components.



- The public server is an Apache-based cache of Sun Update Connection – Enterprise certified components and their deployment rules, along with signature files and other security measures.
- The Sun Update Connection – Enterprise server pulls data, in the form of Apache pages, from the public server and pushes on-demand to agents, consoles, CLI, and API.
- The DM is the Sun Update Connection – Enterprise application that manages jobs. It manages communications between agents, consoles, CLI, and API.
- The knowledge base (KB) is a repository of downloadable packages and their deployment rules.
- The agent is the Sun Update Connection – Enterprise application that is installed on each host to be managed.
- The console is a graphical user interface.
- The CLI is a command-line interface.

- The API is an optional, customer-built interface.

## Defining Component Communications Details

This section explains how the Sun Update Connection – Enterprise components communicate with each other to fulfill Sun Update Connection – Enterprise functionality. In the following table, wherever agent is mentioned, it is true also of the console, CLI, and API.

**TABLE 6–1** Component Communication Details

| Components | Initiator | Data Flow | Connection Timing | Port & Protocol |
|---|---|---|---|---|
| Sun Update Connection – Enterprise server - public server | Sun Update Connection – Enterprise server | pull from public server | scheduled check for new updates;<br>on demand for agent needs;<br>closed when task done. | 443 and HTTPS<br>(443 is saved for HTTPS only) |
| Sun Update Connection – Enterprise server - agents | agents | pull from KB via server | on demand; closed when done | HTTPS<br>8002 (8000-8100) |
| DM - agents | agent - login<br>DM - alive pings<br>both - messages | bi-directional | constant | TCP/TP<br>8100 (8100-8200) DM listening port<br>8200 (8200 - 8300) agent listening for reconnect |
| Sun Update Connection – Enterprise server - DM | none | none | none | none |
| Sun Update Connection – Enterprise server - KB | server | pull from KB to fulfill jobs<br>push of local to KB | on demand; closed when done | cache-type pull |
| server.cgi - LRL | server.cgi | execute LRL | on schedule | application execution |

# Editing the Configuration File

The Sun Update Connection – Enterprise server and the DM are installed with the ezInstaller program.

You can change the configuration of each Sun Update Connection – Enterprise component by editing the .uce.rc file.

**Caution** – Do **not** change the uce.rc file. If you want to make a change, make it in the override .uce.rc file (notice the dot in the override filename). Sun cannot accept responsibility for results if you change the default uce.rc file.

## Configure the Sun Update Connection – Enterprise Server to the Public Server

The Sun Update Connection – Enterprise server is directed to the public server with parameters seen in /usr/local/uce/server/cgi-bin/.uce.rc file:

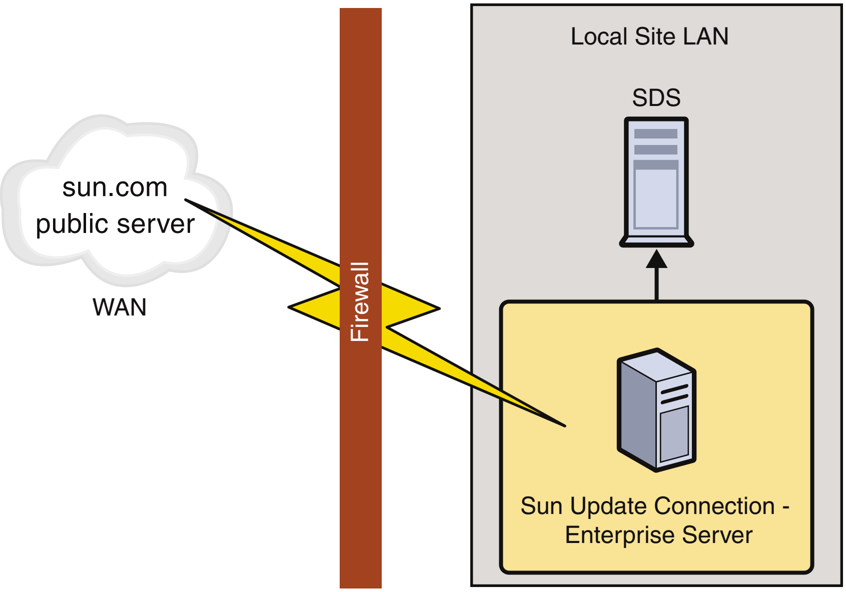
```
( server_name , "knowledge.aduva.com" )
( server_port , 443 )
```

**Caution** – Do **not** change the values of these parameters in either of the rc files. The default value of server_name is the name of the public server, other values do not work. Likewise, server_port 443 is the only value that allows Sun Update Connection – Enterprise to operate correctly.

The Sun Update Connection – Enterprise server is the only Sun Update Connection – Enterprise component that communicates over the Internet (to the public server). The server uses secure HTTPS on port 443.

## Web Proxy and Firewall Connection Parameters

A standard architecture on the local site contains a firewall and, in some cases, a web proxy.



If the Sun Update Connection – Enterprise server is separated from the Internet by a web proxy, configure the connection parameters of the `/usr/local/uce/server/cgi-bin/.uce.rc` file.

Use one of the following methods to configure the connection parameters:

- If your Internet security is in place before you install Sun Update Connection – Enterprise with the `ezInstaller` command, the following message is displayed:

  ```
  Failed to connect to Universal Server.
  If you have a web proxy, configuring it might solve this problem.
  Would you like to setup proxy settings?
  ```

  To setup your proxy settings, type Y and answer the questions to configure your web proxy. When completed, your web proxy parameters are automatically copied to the `.uce.rc` file and the local values are inserted.

- If your Internet security was set up or changed after installation, copy the parameters from the Sun Update Connection – Enterprise server `uce.rc` file to the `.uce.rc` file and change the values for the local environment.

  If you require Web proxy authentication, you might have to create the `/etc/server/.proxyauth`. The `.proxyauth` file contains the following line:

  *username:password*

The following are the web proxy parameters:

```
( all ) ( proxy_server_name , "your_proxy" ); - hostname of web proxy
( all ) ( proxy_server_port , 8080 ); - open port of web proxy
( all ) ( proxy_user_name , "your_authentication_username" ); - username for authentication
( all ) ( proxy_user_password , "your_authentication_password" ); - password for authentication
```

Verify that the `proxy.conf` file is in the `/etc/server/` directory. Web proxy authentication is optional. If you use authentication, the `.proxyauth` file must also be in the `/etc/server/` directory.

# Reconfiguring Agent, Console, CLI, and APIParameters

Agents, consoles, CLI, and APIs are each installed with a separate Install program. During automatic installation, the `.uce.rc` file for each is created and in it are placed local values for these parameters:

```
( server_name , "<server_hostname>" );
( server_port , 8002 );
( distrizor_host , "<engine_hostname>");
( distizor_port , 8100);
( agent.__general.agent_port , 8200 );
```

The parameter files are:

- `/opt/local/uce/agent/bin/.uce.rc` file – on every managed host
- `/~/.uce_console/bin/.uce.rc` file – on `/home` directory of every console user
- `/usr/local/uce/cli/bin/.uce.rc` file – on system with CLI
- `/~/.uce_python/bin/.uce.rc` file – on `/home` directory of the uce-python API user

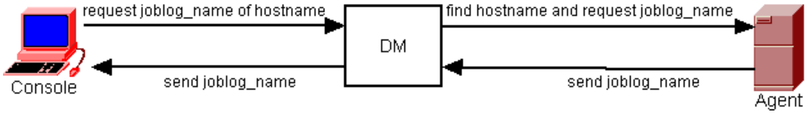
## Agent Connections to Sun Update Connection – Enterprise Server

The default port between agents (including consoles, CLI, and API) and the Sun Update Connection – Enterprise server is 8002. This is the `server_port` parameter. If this port is not available on the local site, the range from 8000 to 8100 is searched. Agents call to the Sun Update Connection – Enterprise server using HTTPS.

If you want a different default port, open the `UCE-<version>-<release>/scripts/defs` file and change the value of `DEFAULT_START_PORT` before installing.

## Agents to DM

Agents, consoles, CLI, and API do not communicate with each other; all their inter-connections are through the DM.

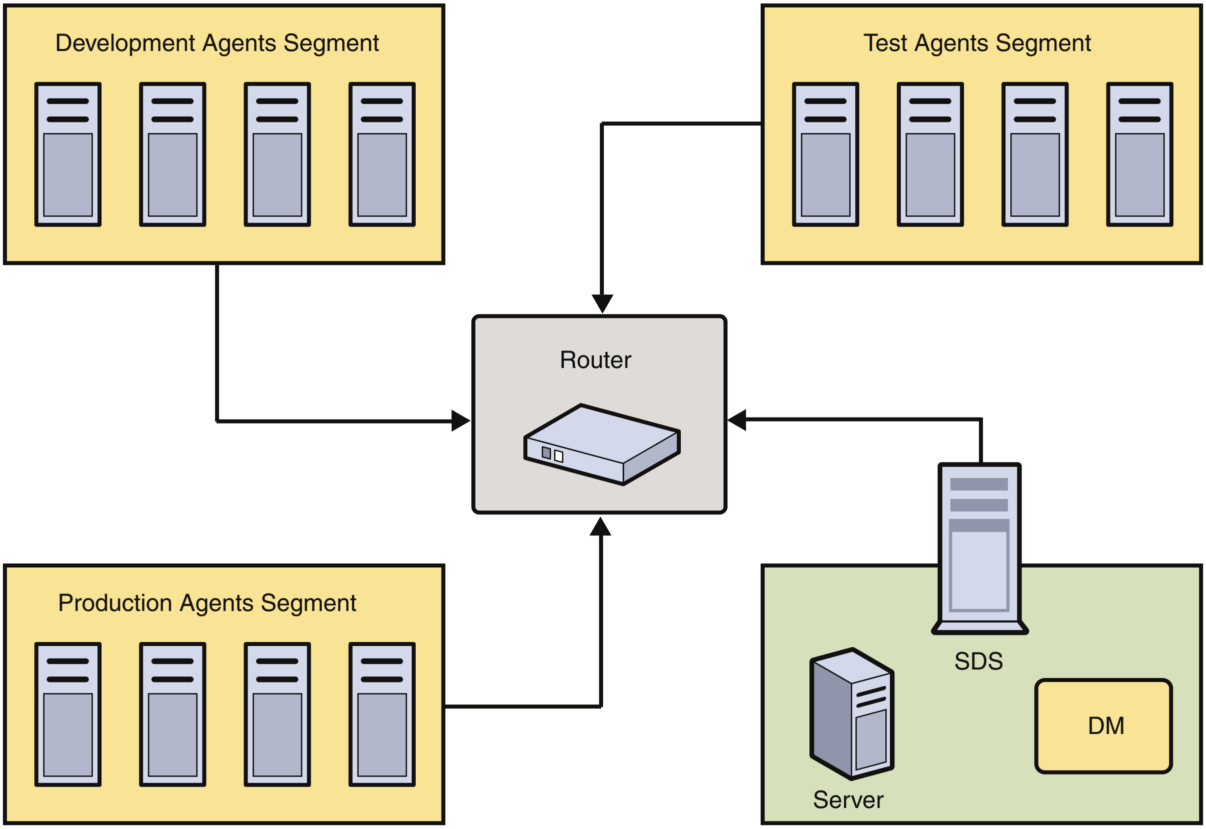

request joblog_name of hostname
send joblog_name
Console
DM
find hostname and request joblog_name
send joblog_name
Agent

Agents (including consoles, CLI, and API) and the DM communicate over TCP/IP. The default listening port for the DM from the agents is 8100. However, when the agent and the DM first communicate, any open port between 8100 and 8200 can become their regular communication port. This port is configured in the `distrizor_port` parameter.

The agent might be released from this connection if traffic to the DM reaches the threshold. The DM maintains a pinging schedule and renew the connection when traffic opens. Agents listen to the DM for reconnection on the agent listening port: 8200. This is the `agent_port` parameter. If the local environment does not allow for 8200 to be used, the scanned range (during installation) is 8200 to 8300.

## Agents on Different Networks

A standard site configuration separates agents by network segments (for example, Development might be a separate segment from Testing), and the agents might be on different networks than the Sun Update Connection – Enterprise server and the DM.

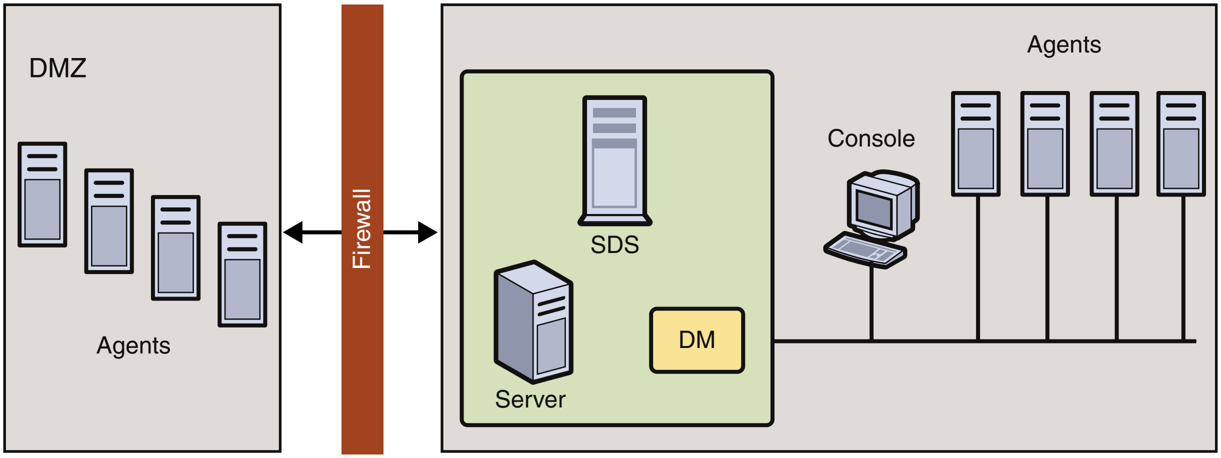


If this is the case, make sure that the ports for `distrizor_port`, `agent_port`, and `server_port` are open.

## Agents With DMZ

If agents are separated from the Sun Update Connection – Enterprise server and DM by a DMZ setup, the agent's `uce.rc` file contains the proxy parameters (as shown in "Agent Connections to Sun Update Connection – Enterprise Server" on page 44) and can be configured in the `.uce.rc` file to reach the Sun Update Connection – Enterprise server.

It is recommended that the Sun Update Connection – Enterprise server and the DM be on the secured LAN, not on the DMZ.


DMZ
Agents
Firewall
SDS
Server
DM
Console
Agents

CHAPTER 7

# Optimizing Performance For Linux

This chapter explains the principles of the technology that guides the performance of Sun Update Connection – Enterprise and how to optimize its features.

- "Defining Linux Knowledge" on page 49
- "Using Local Updates" on page 52
- "Servers in the Solution" on page 53

## Defining Linux Knowledge

The system dependency server is the heart of Sun Update Connection – Enterprise. It contains the dependency manager and the knowledge base.

Your local knowledge base, pulled from the universal server plus local software and data that you pushed from your hosts (to the local knowledge base, never to the Universal Server on the Internet), contains the following Linux information:

- Linux *components* – Logical units that are, or can be, part of a system. For example, RPMs, drivers, modules, and symbols.
- Dependency *rules* – Instructions for successful installation of a component and all of its dependent components.

  The dependency manager, or *Engine*, manages the agents. The Sun Update Connection – Enterprise agent is installed on every managed host and runs the Dependency Resolver, the application that finds the most cost-effective solution for jobs. In this way, every managed host finds the best solution for its own software configuration. Agents solve jobs with rules and components.

### RPM Components

A component is anything that can be downloaded and installed on a system. Most of the certified components on the universal server are RPMs.

Linux is not developed by a single company. Instead, thousands of independent, Open Source developers constantly update and improve the Linux operating system. There are tens of thousands of components available for download and installation.

The RPM Package Manager brings some order to the chaos. The idea of a Linux package provides a standard for independent developers to offer their applications for distribution. The RPM standard describes how to name a package, how to structure its format for consistent management, and the type of data it should or may contain.

A set of components packed with the RPM engine is also called an RPM and is also packed with the following data:

- Date, time, version, release, and environment (such as Intel or zSeries)
- Description of the packed components, including the following: names, size, installation paths, permissions, owners, general security checks, and file contents.
- Reference to other components. For example, what it is similar to, what it is dependent upon, and what it conflicts with.
- a signature to verify authenticity

### ▼ To Create an RPM

1. **Collect the source files of the application and any new updates.**
2. **Write the specification file containing packaging scripts run by the RPM engine and a file list.**
3. **Run the RPM Engine with flags to build the package.**

   Notice that the RPM procedure is rather loose. The developer can decide which source files to include in the package. Other files, such as libraries and basic operating system components, may be simply pointed to in the reference data as dependent packages. This means that the end user must make sure that all packages that the developer assumed were already installed are, in fact, on the system.

   When the user installs an RPM, the scripts (written by the developer, not created by an automated standard) are run and tell the user the following information:

   - If dependent components need to be installed
   - If there are installed components that conflict with files in the package
   - The version range of dependent components and conflicting components

## Dependency Rules

Most Linux components depend upon the prior installation of existing libraries or other packages to operate in known system configurations. These other components are *dependent components*. For example, `kdesdk*.rpm` needs `gcc-c++` to be installed.

When components need incompatible versions of dependent components, a *dependency conflict* arises (packages or files that should not exist on the same system as the base component, such as an earlier version). For example, if `mysql-devel` is installed, `MySQL-devel` does not work.

Descriptions of dependencies and dependency conflicts are called rules.

To install a component manually, without using Sun Update Connection – Enterprise, you must discover and install the complete dependent component list before you can install or run the original component. When you install a component with Sun Update Connection – Enterprise, dependency issues are taken care of automatically. In addition, the rules of the knowledge base are exact and accurate, while the rules included in an Open Source package may not be.

## Standard Dependency Rules

Open Source packaging rules are based on loose standards.

Package developers might write very specific dependency rules, but the rules might be more restrictive than necessary. For example, a specific set of rules might say that you need at least version 1.5 of a software package. You have version 1.3 of the software and upgrade to a version that meets the dependency rules. However,v it could be that earlier versions are not in the rules because the developer did not test those versions.

Other developers might be to general when writing the dependency rules, and you might encounter conflicts even if you follow the rules.For example, a general set of rules may say that you need version 1 to 3 of a software package. Another developer creates v2.9, which causes a conflict with your other installed components

## Sun Update Connection – Enterprise Dependency Rules

The dependency rules in the knowledge base are exact, neither too specific nor too general.

For example, the standard rules for SuSE's `dia-0.85-34` package (an application for creating diagrams and flowcharts) say that this package needs `libxml` of any version. Among the certification tests, the `dia` package was installed on a system with `libxml-1.7.3-3`. An installation error resulted. Running more tests, the exact Sun Update Connection – Enterprise rule was created: `dia-0.85-34` needs `libxml` from version `1.8.6-18`.

The Certification Lab tests every component and finds its *exact* installation rules, listing, including the following:

- All dependent components
- All component conflicts

  In addition, if a dependent component has its own dependencies, this is also known. There is no limit to the number of levels that Sun Update Connection – Enterprise takes into account.

  The knowledge base contains the dependency rules, and Sun Update Connection – Enterprise uses it to automate functional installations. Sun Update Connection – Enterprise also finds potential conflicts, warns you of them, and suggests alternatives before you install a component.

# Using Local Updates

The knowledge base is updated continuously, providing you with relevant update management from the Linux community and public update releases.

In addition, you may have private components that your own organization has patched and customized. Using Sun Update Connection – Enterprise, you can mark a local component as a security fix for a previous local component and upload the security fix to the local knowledge base.

Marking a local component as a security fix has the following results:

- During the predefined profile **Security Check**, if a host has the first version installed, it is upgraded to the Security Fix version.
- During a user-defined profile, if the job should install the local package as a dependency, the Security Fix version is given priority over the earlier version. If the job is marked for **Use secure components only**, only the Security Fix version can be installed. The job fails instead of installing the unsecured version.

  If a later update is created that provides enhancements on the Security Fix package, you may decide that both local packages are Security Fixes, or that only the latest package is the one preferred Fix. From the console , you can select a Local Software component and then change its Security Fix mark.

## ▼ To Mark a Local RPM as a Fix

**Before You Begin** Ensure that the Inventory panel is visible. From the View menu, choose Inventory.

1. **Log in to the system as a superuser.**

2. **From the drop-down list in the tool bar, select a distribution-architecture.**

   The Components list changes to show the components relevant to your selection. The NCOs that you add with this procedure are added to the inventory of the displayed distribution.

3. **Under the Local category, select Local RPMs or a user-defined category under it.**

4. **Right-click the selected category and choose Local, then choose Add.**

   The Add Software window opens.

5. **Select whether the RPM is accessed from the localhost (console) or from a remote managed host.**

6. **Browse to the file if it is on the console; type in the path name if it is on a remote host.**

7. **Check Security Fix.**

**8 Click OK.**

The Add Package window closes. The software is added to the knowledge base as an update to the previous like-named added components.

## Secure Components Default Setting

The job option Use secure components only, which makes sure that all dependencies installed for a job do not have later uninstalled updates, makes the job run slower and take up more resources; therefore, it is deselected by default. If you are sure that you want all jobs to run this check before installing anything, you can change the default settings.

### ▼ To Change Console Preferences for Secure Components

**1 From the Tools menu, choose Preferences.**

The Preferences window opens.

**2 Select the Console radio button.**

**3 In the Category list, choose Jobs.**

**4 Select the Check security checkbox.**

**5 Click Submit.**

The Preferences Confirmation window opens.

**6 Click Submit.**

You do not need to logout to apply this setting.

**7 Open the New Job window, Options tab.**

Notice that Use secure components only is selected by default.

# Servers in the Solution

When an agent registers with the dependency manager (DM), it downloads the rules and the components from the knowledge base. The system dependency server verifies that the knowledge base is updated from the universal server, and in turn, the DM updates the agents. The DM also updates the console, to give you an easy-to-read hierarchy of available components.

In a simplified explanation, the applications work together as follows:

1. Agents and consoles download the rules and components from the knowledge base.
2. After you create a job on the console, the dependency manager picks up the job and distributes it to the selected agents.

3. Each selected agent runs the dependency resolver that uses the rules to find the most cost-efficient solution for its own managed host.
4. The agents use certified components to complete the job.
5. The job results are sent to the dependency manager, which updates the console and the log database.

## Server Security

Sun Update Connection – Enterprise has a variety of security measures integrated into its solution.

**TABLE 7–1** Integrated Server Security

| | |
|---|---|
| Key Encryption | Communications between the dependency manager and the agents and console are protected by a private/public key encryption algorithm. |
| Proxy Support | Sun Update Connection – Enterprise supports HTTPS web proxy servers, enabling another level of security to updates from the universal server. |
| Internet Protection | All Internet communications are protected by SSL and certificate verification.<br>All communications from the agents are proactive; the system dependency server pulls only from the universal server, never pushes data to it. |
| Component Security | Every component available for download from the universal server is certified. You cannot download Trojan horses or other exploits that are disguised as useful software. |

◆ ◆ ◆ CHAPTER 8

# Backup and Restore

The backup and restore tools allow you to backup the system dependency server data files and to restore them at a later date. It provides a disaster recovery solution, for the possibility of the system dependency server system breaking down.

The following topics are covered:

## Server Backup and Restore

Schedule a daily run of the backup script, and save the resulting archive in a secure magnetic media external from the SDS system. A faithful backup procedure allows for full disaster recovery.

### Backing Up the System Dependency Server

This task explains what happens when you run the backup script. The `backup.sh` script is in the installation directory created when you first installed the system dependency server: `/usr/local/uce/install`.

#### ▼ To Back Up the System Dependency Server

1. **Log in to the system as superuser.**

2. **Change to the installation directory.**

```
# cd /usr/local/uce/install
```

**3 Run the backup script.**

```
# ./backup.sh
```

## Backup Archive

The following steps are performed automatically:

1. Sun Update Connection – Enterprise applications are stopped.
2. The backup script verifies that the local system is the system dependency server.
3. The data in the following directories and files are saved under backup names:
   - `/usr/local/uce/server/public/*` - Universal rules and components
   - `/usr/local/uce/server/private/*` - Local rules and components
   - `/usr/local/uce/engine/*/bin` - Settings files and encryption keys
   - `/usr/local/uce/install/*` - Application installers, restore and support scripts
4. The databases are backed up and the backup files are stored in a tar file. The file name includes the year, month, day, hour, and minute that the backup was created.`/usr/local/uce_backup/backup-<yyyy-mm-dd-hh-mm>.tar.gz`
5. Sun Update Connection – Enterprise applications are restarted and backup exits.

**EXAMPLE 8–1** Backup Archive File

The following backup archive contains the contains local and universal knowledge base and console data for June 25, 2003, 1:07 p.m.

```
/usr/local/uce_backup/backup-2003-06-25-13-07.tar.gz
```

# Restoring the System Dependency Server

This task explains what happens when you run the restore script. The `restore.sh` script is in the `/usr/local/uce/install` directory.

**Caution** – Once you select a file to restore, all current Sun Update Connection – Enterprise data is overwritten. Before restoring an older version, you should perform the backup procedure for the current data.

## ▼ To Restore the System Dependency Server

**1 Log in to the system as superuser.**

**2 Change to the installation directory.**

```
# cd /usr/local/uce/install
```

3 **Run the restore script with a pointer to the desired backup archive.**

```
# ./restore.sh /<fullpath>/<backedUpServer>.tar.gz
```

The following steps are done automatically, but require your confirmation before continuing:

- Sun Update Connection – Enterprise applications are stopped.
- The selected file is unpacked and the data is restored.
- Sun Update Connection – Enterprise applications are restarted and restore exits.

# Shared Resources Server Backup and Restore

If managed hosts use shared resources, perform the following tasks in addition to the previous procedures. Unless otherwise stated, do the tasks on the system dependency server system.

## ▼ To Back Up Files for Shared Resources

1 **Change to the install directory.**

```
# cd /usr/local/uce/install
```

2 **Create the console subdirectory.**

```
mkdir console
```

3 **Create the agent subdirectory.**

```
mkdir agent
```

4 **From the console system, copy the console's `.uce.rc` file to the new console directory on the system dependency server system.**

```
cp ~/.uce_console/.uce.rc /usr/local/uce/install/console
```

5 **From a managed host, copy an agent's `.uce.rc` file to the new agent directory on the system dependency server.**

```
cp /opt/uce/agent/bin/.uce.rc /usr/local/uce/install/agent
```

6 **Run the backup script.**

```
./backup.sh
```

The `.uce.rc` files of the consoles and agents contain the parameter that enables shared resources support. These parameters are included in the backup.

### ▼ To Restore Shared Resource Files

1. **Execute the restore script.**

   ```
   # ./restore.sh
   ```

2. **Change to the installation directory.**

   ```
   # cd /usr/local/uce/install
   ```

3. **If the agents or consoles were changed after backup, reinstall them using the new `console-date.tar.gz` and `agent-date.tar.gz` files. Do not open the console or start agents.**

4. **Overwrite the newly installed `.uce.rc` file of the console with the file that was backed up.**

   ```
   cp /usr/local/uce/install/console/.uce.rc ~/.uce_console/.uce.rc
   ```

5. **Stop the agents. On each managed host.**

   ```
   # /etc/init.d/uce_agent stop
   ```

6. **Overwrite the newly installed .uce.rc file of the agent on each managed host: `/opt/uce/agent/bin/.uce.rc` with the file backed up on the SDS: `/usr/local/uce/install/agent/.uce.rc`**

7. **On each managed host, start the agents.**

   ```
   # /etc/init.d/uce_agent start
   ```

   All affected systems are restored to their exact condition of the time of the backup.

# Disaster Recovery

Follow the steps of this procedure to use the backup and restore tools to recover from a complete failure of the system dependency server system.

## Recovering the system dependency server

This task is most effective if you have made daily backups and stored them externally.

### ▼ To Recover From a System Dependency Server Disaster

1. **Perform a daily backup as described previously.**

2. **Select a backup server to replace the broken server. Change the hostname of the new server to match the name of the faulty system.**

3. **Change DNS to resolve the hostname to the new server's IP address.**

4. **Install the system dependency server on the new system.**

5. **Restore the last backup archive by using the `restore.sh` script.**

6. **Start the console.**

7. **Manually apply all unsaved changes made to the system dependency server after the last backup.**

# Troubleshooting

- If the script seems to take some time before starting, Sun Update Connection – Enterprise is busy. The backup or restore tool waits for the procedure to be completed before continuing.
- If the `ezInstaller` command is run at any time after backup and before restore, you must reinstall the agents and console from the `/usr/local/aduva/install` directory. This is to ensure that the system dependency server, the agents, and the console all have the same encryption keys. Without the same keys, communication between the engine, console and agents is not possible.
- If an error appears that the database dump is empty and then the script exits, contact technical support.
- If the backup script exits without backing up, the local system is not the system dependency server. This script backs up only the system dependency server, not agents or consoles.

# Index