



Sun N1 Service Provisioning System 5.1 Operation and Provisioning Guide

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

Part No: 819-1658-10
September 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



050826@12762



Contents

Preface	7
1 Deploying Components	11
About Component Deployment	11
Using Direct-Run Component Procedures	12
▼ How to Use Direct-Run Component Procedures	12
Using Plans	15
▼ How to Create Generated Plans	15
Managing Component Deployment	17
▼ How to Run a Plan	17
▼ How to View the Progress of a Plan	19
▼ How to Stop a Running Plan	20
▼ How to Delete a Plan	20
Deployment Examples	21
Updating a Component	21
Downgrading to a Previous Component Version	22
▼ How to Downgrade to a Previous Component Version	22
Migrating a Component From One Server to Another	23
Grouping Simple Components in Container Components	23
▼ How to Create a Container Component	23
2 Running Comparisons	25
About Comparisons	25
Time Synchronization for Comparisons	26
Model to Model Comparisons	27
▼ How to Create a Model to Model Comparison	27

Model to Install Comparisons	28
Component Types and Model to Install Comparisons	30
▼ How to Create a Model to Install Comparison	32
Install to Install Comparisons	34
Using Scripts in Install to Install Comparisons	35
▼ How to Create an Install to Install Comparison	36
Managing Comparisons	38
▼ How to Run a Comparison	38
▼ How to View Comparisons	39
▼ How to View the Progress of a Comparison	40
▼ How to Stop a Running Comparison	40
▼ Modifying Directories or Files to Ignore During Comparisons	41

A Troubleshooting Deployment Issues	43
Error Message and Descriptions	43
Log Messages and Descriptions	46
Memory Issues	46

Glossary	49
-----------------	-----------

Index	57
--------------	-----------

Examples

EXAMPLE 2-1	A <code>prepare</code> Block for an Install to Install Comparison	35
EXAMPLE 2-2	A <code>cleanup</code> Block for an Install to Install Comparison	35

Preface

The *Sun N1 Service Provisioning System 5.1 Operation and Provisioning Guide* provides procedures for deploying components and comparing hosts, host sets, and components.

Who Should Use This Book

This book is for users of the N1 Service Provisioning System software who are responsible for running plans and ensuring that systems in their environment have the correct resources installed.

How This Book Is Organized

[Chapter 1](#) describes the different mechanisms available to deploy resources to hosts and host sets.

[Chapter 2](#) describes how to create and run the three different types of comparisons.

Related Books

The following books belong to the Sun N1 Service Provisioning System 5.1 documentation collection. All books listed below are available on docs.sun.com (<http://docs.sun.com>).

TABLE P-1 Sun N1 Service Provisioning System 5.1 Documentation

Application	Title
Software Installation	<i>Sun N1 Service Provisioning System 5.1 Installation Guide</i>
System Administration	<i>Sun N1 Service Provisioning System 5.1 System Administration Guide</i>
Plan and component creation	<i>Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide</i>
Command reference	<i>Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual</i>
XML Schema Reference	<i>Sun N1 Service Provisioning System 5.1 XML Schema Reference Guide</i>
Release Notes	<i>Sun N1 Service Provisioning System 5.1 Release Notes</i>
Plug-in Development	<i>Sun N1 Service Provisioning System 5.1 Plug-in Development Guide</i>

Documentation, Support, and Training

Sun Function	URL	Description
Documentation	http://www.sun.com/documentation/	Download PDF and HTML documents, and order printed documents
Support and Training	http://www.sun.com/supporttraining/	Obtain technical support, download patches, and learn about Sun courses

Typographic Conventions

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

TABLE P-2 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name%</code> su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . Perform a <i>patch analysis</i> . Do <i>not</i> save the file. [Note that some emphasized items appear bold online.]

Shell Prompts in Command Examples

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

TABLE P-3 Shell Prompts

Shell	Prompt
C shell prompt	<code>machine_name%</code>
C shell superuser prompt	<code>machine_name#</code>
Bourne shell and Korn shell prompt	<code>\$</code>

TABLE P-3 Shell Prompts *(Continued)*

Shell	Prompt
Bourne shell and Korn shell superuser prompt	#

Deploying Components

This chapter provides a brief description of what plans are and how to deploy components.

About Component Deployment

You can deploy components using any of three methods.

- Component procedures. For simple, single operations involving one component and one host or host set.
- Plans
 - Generated plans. For multiple operations involving one component and one host or host set.
 - Custom plans. For complex operations involving multiple components and multiple hosts.

The provisioning system can deploy any number of components to a valid host. The number of components that can be deployed is only limited by the file system on the host. Each file system limits the number of files that can be deployed, and by deploying too many components to a single directory, you can exceed the file systems limitations. The relationship between components and files is not necessarily an equal ratio. For example, one component might exceed file system limitations if it has 100,000 files, and 100,000 components might fall within the file system limitations if none of the components contain files.

Exceeding file system limits might not result in a direct failure when the components are deployed. However, problems might arise in the form of increasingly severe performance degradation or unpredictable failures at other times.

Limits to the number of files that a directory can support vary by file system and can be influenced by how the operating system is configured. You can avoid overloading a directory by understanding your file system's limitations and deploying components into smaller subdirectories, rather than into a single, large directory. A conservative practice to use when deploying components is to place no more than 30,000 files in a single directory.

Using Direct-Run Component Procedures

For straightforward deployments that don't involve multiple components or synchronization among hosts, you might not need a plan to deploy. Instead, you might be able to use one of the direct-run component procedures that the provisioning software automatically generates when you check in a component that you want to deploy.

▼ How to Use Direct-Run Component Procedures

Most components include install, uninstall and snapshot procedures. A component might also include control procedures, which manage the installed component. For example, a control procedure might start or stop the application.

This procedure describes how to deploy a component directly by using the browser interface. Use the following commands to perform the equivalent functionality.

- `pdp.p.genplan` – Generates a plan.
- `pe.p.run` – Runs a plan.

For a detailed description of this command, see Chapter 9, “pdb: CLI Commands for Managing Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual* and Chapter 10, “pe: CLI Commands for Running Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

For more information on component development, see *Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide*.

Note – There is no need to delete the plans generated from direct-run component procedures. These plans are automatically removed from the system when the component from which they were generated is deleted. For more information, see “How to Delete a Component” in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide*.

Before You Begin

To use direct-run component procedures, you must belong to a user group that has Run Component Procedures permission on the folder that contains the component. You must also have Allow on Host Set permission for the host set on which you plan to deploy the component.

- Steps**
1. **From the navigation menu, choose Components.**
The Components page is displayed.
 2. **In the table listing components, find the row describing the component you would like to deploy and click Details.**
The component's Details page is displayed.
 3. **If necessary, change folders so that you can view the component that you plan to deploy.**
 - a. **Click Change Folder.**
 - b. **Specify the name of the folder that contains the component, and click Change to Selected Folder.**
The Components page now lists the components in the specified folder.
 4. **In the Component Procedures table, determine the procedure that you want to run and click Run.**
The provisioning system generates a plan, which is filed in the `/system/autogen` folder.
The Run page for the generated plan displays.
 5. **In the Plan Parameters area, select the variable settings for the component you plan to deploy.**
 - **If the variable settings have been established for this component, select the appropriate settings from the menu.**
 - **If the settings are not available from the menu, click Select From List.**
The Select Variable Settings From List window displays.
 - **If you want to use another component's variable settings, click Import Set.**
The Import Variable Settings window displays.
 - a. **If necessary, navigate to the Folder that contains the component with the variable settings you want to import.**
 - b. **Select the component version.**

Note – Variable settings can vary between component versions.

- c. **Click Import Variable Settings.**
The imported variable settings display in the table.
- d. **Click Select.**
- **If you want to create new variable settings, select Create Set.**

The Select Variable Settings From List window displays.

- a. **In the text field at the top of the table, enter the name of the new variable settings set.**
- b. **Select the component variable you want to change.**
The table cell is highlighted and a text field appears.
- c. **Type the new value of the component variable.**
- d. **After updating the variable settings values, click Save.**
The new variable settings display in the table.
- e. **Click Select.**

6. **Select where you plan to deploy the component.**

- **To select a host, use the Target Host text field to type a host's name or click Select From List.**
- **To select a target host set, use the Target Host Set menu.**

7. **In the Plan Variables area, configure the variables you want to set.**

Whether there are variables to configure depends on the contents of the component procedure you select.

8. **In the Options area, select whether you want to perform a detailed preflight.**

The detailed preflight attempts to simulate every step of the plan that will run on the remote agent. The detailed preflight tests the following functions.

- Resource installation
- `execNative` calls
- File transformations
- File deletion, when uninstalling resources

Running a detailed preflight increases the amount of time preflight takes.

9. **(Optional) To limit the number of hosts running at the same time, type the number of hosts in the field provided.**

When a plan is run against several hosts simultaneously, all hosts must complete a step before any host can progress to the next step. If the number of hosts is too high, the network connection can time out between steps. For example, the time between Host A finishing step 1 and being pinged for step 2 might be too long and cause the network connection to time out.

10. **(Optional) To limit overall running time of a plan or to limit the running time of native calls, type a number in the fields provided. Then select the time unit from the menu.**

By limiting the run time of a plan or native call, you prevent a nonresponsive host from tying up the progress of a plan run.

If you set up a notification rule to alert you to plan failures, when the plan times out you will be notified that the plan was unsuccessful.

For information on setting up notification rules, see “Configuring Email Notification” in *Sun N1 Service Provisioning System 5.1 System Administration Guide*.

11. Select whether you want to deploy the component.

- To run just the preflight test of the procedure, click **Run Preflight Only**.
- To run the preflight test and then (assuming the preflight test finds no errors) to run the procedure itself, click **Run Plan (Includes Preflight)**.

Using Plans

You can run two types of plans in the provisioning system.

- Generated plans
- Custom plans

Generated Plans

The provisioning system can automatically generate a plan consisting of multiple component procedures. You can run this plan directly or save it for use as a template for more sophisticated plans that you author in XML.

Custom Plans

You can author custom plans in XML. For information about writing custom plans, see “Creating Plans” in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer’s Guide* and the *Sun N1 Service Provisioning System 5.1 XML Schema Reference Guide*.

▼ How to Create Generated Plans

This procedure describes how to generate a plan by using the browser interface. You can also generate plans by using the following commands.

- `pdp.p.genplan` – Generates a plan.
- `pe.p.run` – Runs a plan.

For a detailed description of these commands, see Chapter 9, “pdb: CLI Commands for Managing Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin To create a generated plan, you must belong to a user group that has Create, Edit, Delete permission on the folder that contains the component for which you want to generate a plan.

- Steps**
1. **From the navigation menu, choose Components.**
The Components page displays.
 2. **If necessary, change folders so that you can view the component that you plan to deploy.**
 - a. **Click Change Folder.**
 - b. **Specify the name of the folder that contains the component, and click Change to Selected Folder.**
The Components page now lists the components in the specified folder.
 3. **In the table listing components, find the row describing the component you would like to deploy, and click Details.**
The component's Details page displays.
 4. **In the Component Procedures table, select each procedure that you want to include in the plan.**
 5. **In the bottom row of the Component Procedures table, click Generate Plan With Checked Procedures.**
The generated plan's Advanced Edit displays.
 6. **In the Plan text field, type the new plan's name.**
 7. **Click Check In.**

Note – You might see this message: Warning - plan names and/or paths differ. This message indicates that you specified a plan name different from the plan name specified in the XML model. The XML model is updated according to what you type into the plan name text field.

8. **Confirm the component check in.**
 - a. **Verify that you have specified the correct plan name.**
 - b. **If necessary, select the component's new version number.**
 - c. **Click Continue to Check In.**
The new plan's Detail's page is displayed.

Managing Component Deployment

Deploying components with the help of plans can often take a significant amount of time. To provide visibility into the process of deploying components, you can monitor plans and view information about running or completed plans. You can also stop a plan in the middle of a deployment.

- [“How to Run a Plan” on page 17](#)
- [“How to View the Progress of a Plan” on page 19](#)
- [“How to Stop a Running Plan” on page 20](#)

▼ How to Run a Plan

This procedure describes how to run a plan by using the browser interface. You can also run plans by using the following command.

```
pe.p.run – Runs a plan.
```

For a detailed description of this command, see Chapter 10, “pe: CLI Commands for Running Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin Ensure that you have the installation information that will allow you to set the plan’s variable settings.

To run a plan, you must belong to a user group that has `Allow on Host Set` permissions in the folder that contains the plan you need to run.

- Steps**
1. **Go to the plan’s Details page.**
 2. **Click Run.**
The plan’s Run page displays.
 3. **In the Plan Parameters area, select the variable settings for the component you plan to deploy.**
 - **If the variable settings have been established for this component, select the appropriate settings from the menu.**
 - **If the settings are not available from the menu, click Select From List.**
The Select Variable Settings From List window displays.
 - **If you want to use another component’s variable settings, click Import Set.**
The Import Variable Settings window displays.

- a. If necessary, navigate to the Folder that contains the component with the variable settings you want to import.
- b. Select the component version.

Note – Variable settings can vary between component versions.

- c. Click **Import Variable Settings**.
The imported variable settings display in the table.

- d. Click **Select**.

- If you want to create new variable settings, select **Create Set**.
The Select Variable Settings From List window displays.

- a. In the text field at the top of the table, enter the name of the new variable settings set.

- b. Select the component variable you want to change.
The table cell is highlighted and a text field appears.

- c. Type the new component variable value.

- d. After updating the variable settings values, click **Save**.
The new variable settings display in the table.

- e. Click **Select**.

4. Select where you plan to deploy the component.

- To select a host, use the **Target Host** text field to type a host's name or click **Select from List**.
- To select a target host set, use the **Target Host Set** menu.

5. In the **Plan Variables** area, configure the variables you want to set.

Whether there are variables to configure depends on the contents of the plan.

6. In the **Options** area, select whether you want to perform a detailed preflight.

The detailed preflight attempts to simulate every step of the plan that will run on the remote agent. The detailed preflight tests the following functions.

- Resource installation
- `execNative` calls
- File transforms
- File deletion, when uninstalling resources

Running a detailed preflight increases the amount of time preflight takes.

7. (Optional) To limit the number of hosts running at the same time, type the number of hosts in the field provided.

When a plan is run against several hosts simultaneously, all hosts must complete a step before any host can progress to the next step. If the number of hosts is too high, the network connection can time out between steps. For example, the time between Host A finishing step 1 and being pinged for step 2 might be too long and cause the network connection to time out.

8. (Optional) To limit the overall running time of a plan or limit the running time of native calls, type a number in the fields provided, and then select the time unit, such as minutes, from the menu.

By limiting the run time of a plan or native call, you prevent a non-responsive host from tying up the progress of a running a plan.

If you set up a notification rule to alert you to plan failures, when the plan times out you will be notified that the plan was unsuccessful.

For information on setting up notification rules, see “Configuring Email Notification” in *Sun N1 Service Provisioning System 5.1 System Administration Guide*.

9. Select whether you want to deploy the component.

- To run just the preflight test of the procedure, click **Run Preflight Only**.
- To run the preflight test and then (assuming the preflight test finds no errors) to run the procedure itself, click **Run Plan (Includes Preflight)**.

▼ How to View the Progress of a Plan

This procedure describes how to view the progress of a plan by using the browser interface. You can also view a plan’s history by using the following commands.

- `pe.p.la` – Lists running and completed plans.
- `pe.p.lo` – Lists information about a running or completed plan.
- `pe.pi.lo` – Lists the parameters used to run a plan.

For a detailed description of these commands, see Chapter 10, “pe: CLI Commands for Running Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

- Steps**
1. In the navigation menu, expand the **Application Deployment** category and click **Run History**.

All running and completed plans and preflight checks are listed. A running plan is indicated by a timestamp in the Completed column. The timestamp is the time at which the plan run was initiated.

2. To view the details about a particular plan, click **Results**.

By selecting the Details, Preflight, or Deployment icons, you can toggle between the plan details, the results of the preflight, and the results of the deployment.

The Details page shows the following information.

- Target host
- Variable settings used for the plan's parameters
- Limits imposed on the plan
- User who ran the plan

▼ How to Stop a Running Plan

This procedure describes how to stop a plan by using the browser interface. You can also stop plans by using the following command.

`pe.p.stop` – Stops a running plan.

For a detailed description of this command, see Chapter 10, “pe: CLI Commands for Running Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin To stop a plan, you must be either the user that started the plan or a member of the admin user group.

Step ● Cancel the running plan.

- **From the Run Deployment page, click Cancel Plan.**
The Run History page displays a message stating that the plan was cancelled by a user.
- **From the Run History page, determine the plan you want to stop.**
 - a. **In the Action column, click Progress.**
The Run Deployment page displays.
 - b. **Click Cancel Plan.**

▼ How to Delete a Plan

This procedure describes how to delete a plan that was not generated from a component by using the browser interface.

Note – There is no need to delete the plans generated from direct-run component procedures. These plans are automatically removed from the system when the component from which they were generated is deleted. For more information, see “How to Delete a Component” in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer’s Guide*.

You can also delete plans by using the following command.

- `pe.p.del` – Deletes a plan.

For a detailed description of this command, see Chapter 10, “pe: CLI Commands for Running Plans,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin To delete a plan, you must either be the user that created the plan or a member of the `admin` user group.

- Steps**
1. **Go to the plan’s Details page.**
 2. **Click Delete.**
You are asked to verify that you want to delete the plan.
 3. **Click Continue to Delete.**
After the plan has been deleted, the Plan table no longer displays the plan.

Deployment Examples

You can use direct-run component procedures and generated plans to manage your applications.

- [“Updating a Component” on page 21](#)
- [“Downgrading to a Previous Component Version” on page 22](#)
- [“Migrating a Component From One Server to Another” on page 23](#)
- [“Grouping Simple Components in Container Components” on page 23](#)

Updating a Component

You can update a component to a newer version of that component. The newer component might reflect a product revision, a change to a file, or a directory that has new files added to it.

If you need to update a simple component, you only need to run the direct run install procedure. The install procedure will overwrite the old component version with the new version.

If you need to update a composite component (a component that references more than one simple component) and the set of referenced components are the same in each component version, you can run the direct run install procedure. The install procedure will overwrite each of the old simple components with the new versions.

If you need to update a composite component and the new version of the component contains a different set of referenced components, you will need to uninstall the old component version and install the new version. By uninstalling the old version, you prevent components that are no longer used or updated from cluttering your work space.

Downgrading to a Previous Component Version

Sometimes the upgrade of a component fails or you experience problems with the upgraded application. In such cases, you can downgrade to a reliable version of the component while you troubleshoot the new component version.

▼ How to Downgrade to a Previous Component Version

Before You Begin To redeploy a reliable component version, you must have Run Component Procedure and Allow on Host Set permissions on the folder that contains the component.

- Steps**
1. **Uninstall the broken component.**
 - a. **From the component's Details page, click Run for the Default: Uninstall component procedure.**
 - b. **From the plan's Run page, select the host from which you want to uninstall the component.**
 - c. **Click Run Select Installations.**
 - d. **Click Run Plan (Includes Preflight).**

By uninstalling the broken component version you ensure that old, unwanted files are removed from the system you deployed to.
 2. **Select the component version that you need to roll back to.**
 - a. **From the component's Details page, click Version History.**
 - b. **From the Version History page, display the reliable component version by clicking Unhide and then click Details.**

3. Install the reliable version of the component.

Migrating a Component From One Server to Another

You can migrate a component from one server to another by simply uninstalling the component you want to move from its current host and reinstalling the component on the new host.

Grouping Simple Components in Container Components

You might want to create a container component if you need to deploy multiple simple components as a group to the same host or host set. Grouping simple components speeds deployment time and ensures that no component is forgotten when the set of components is deployed.

▼ How to Create a Container Component

Before You Begin

To create a container component, you must have Create, Edit, Delete permission on the folder in which you want to create the component.

Steps

1. From the Components page, create a new component.
 - a. In the Component field, type the name of the new component.
 - b. In the Type field, select System#container.
 - c. In the Description field, type a description of the new component.
 - d. Click Create.
2. In the Referenced Components section, add components to the container.
 - If the components you need to reference already exist, click Reference Existing.

The Reference Existing Components in Current Component window appears.

 - a. Select the component that you want to reference in the container component and click Reference Checked Components.

If you reference a component that contains component variables, the component variables are listed in a table before they are added to container component.

b. If necessary, set the component variable's default value in the text field next to the component variable name.

c. Click Continue to Reference Components.

You are returned to the container component's Edit page. The referenced component is displayed in the Referenced Components table.

■ **If the component you need to reference does not exist, click Check In New.**

Create the new component.

For information about how to create a new component, see Chapter 2, "Components," in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide*.

After you have created the new component, the container component's Referenced Components table refreshes with the new component listed.

3. If necessary, provide values for the component variables that need to be defined by the container component.

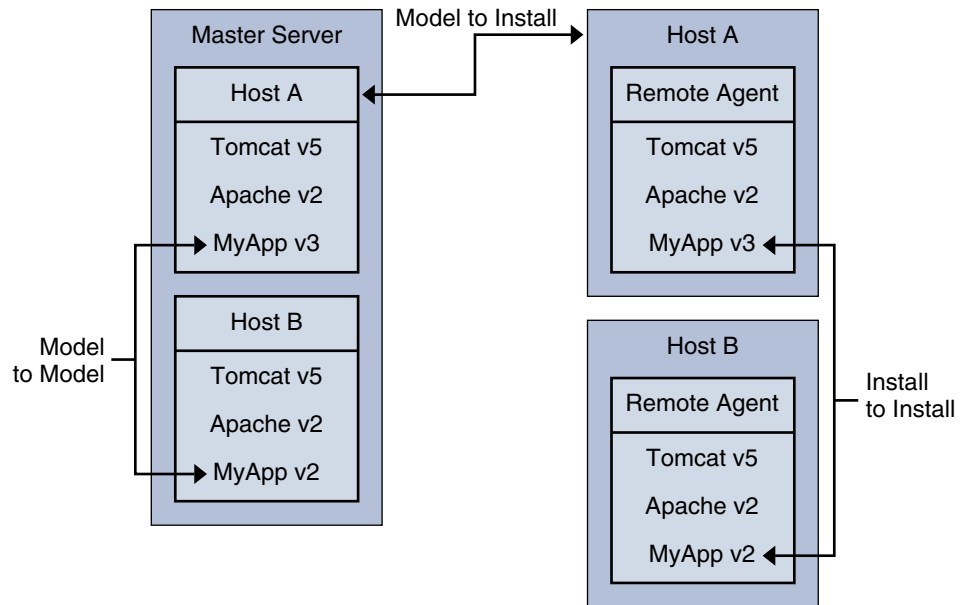
If a component has a `:[container . . .]` substitution variable, the container component needs to define the value.

Running Comparisons

- “Model to Model Comparisons” on page 27
- “Model to Install Comparisons” on page 28
- “Install to Install Comparisons” on page 34
- “Managing Comparisons” on page 38

About Comparisons

Comparisons are used to validate an installation or system configuration. By comparing the configuration of a host against its original configuration, you can tell if anything has changed. You can also compare one or more hosts against a reference host to ensure that all hosts are configured the same.



With the provisioning system you can compare one or more hosts against another host, one or more hosts against a model in the Master Server, or a host against its original configuration. The provisioning software allows you to check the entire configuration of a host or limit what is checked to a single directory.

N1 Service Provisioning System software allows you to perform three types of comparisons.

- Model to Model
- Model to Install
- Install to Install

Time Synchronization for Comparisons

Because the time stamp for a comparison is based on the clock in the Master Server, you should run some kind of synchronization utility on all your N1 Service Provisioning System software machines so that all system clocks are synchronized on the same time.

Model to Model Comparisons

A Model to Model comparison is the simplest type of comparison. The comparison checks the list of components that are installed on one host against the list of components that are installed on another host. This comparison is purely a verification of the records that the provisioning system has about what it has installed.

If there have been out-of-band changes made to either component list, the changes are not detected. A Model to Model comparison takes place entirely on the Master Server based on information the Master Server has received about what has been deployed.

Model to Model Comparison Characteristics

- Compares the records of what has been installed on two hosts
- Requires no data transfer
- Requires the least amount of time to run

You might want to use a Model to Model comparison in the following situations.

- You want to compare the contents of two hosts and no out-of-band changes have occurred.
- You want to compare two hosts and neither of the hosts had snapshots taken during installation.

▼ How to Create a Model to Model Comparison

This procedure describes creating a comparison by using the browser interface. You can also create comparisons by using the following command.

```
cmp . ds . add – Creates a new comparison.
```

For a detailed description of this command, see Chapter 5, “cmp: CLI Commands for Running Comparisons,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin To create a new comparison, you must belong to a user group with Create, Edit, Delete permission on comparisons.

- Steps**
1. **In the navigation menu, click Comparisons.**
The Comparisons page is displayed.
 2. **Type the name and description of the comparison you want to create and click Create.**
The comparison’s Details page is displayed.

3. **In the Style of Comparison area, click Model to Model.**
4. **In the Reference Host field, type or select the name of the host whose model should serve as the standard for the comparison.**
Ideally, the reference host contains the configuration that you would like to see on the comparison host.
5. **In the Comparison Host Field, type or select the name of the host whose model you would like to compare to the reference model.**
6. **In the Limits area, specify the maximum number of hours or minutes the comparison can run.**
By default, comparisons are allowed to run for four hours.
If the comparison times out, an error message will display on the Comparisons page.
7. **Click Save.**

Model to Install Comparisons

Unlike Model to Model comparisons, which compare one ideal installation to another, Model to Install comparisons compare the actual state of the installed component or components at deployment time against their current state. You can specify that this comparison compares the model to hosts, host sets, or to component level.

A Model to Install comparison uses snapshot information to compare a component's just-installed state with its current state. A snapshot is a record of the state of an installed component immediately after it has been deployed to a host. The component's XML model includes parameters that control the scope of this snapshot. For example, the XML model can list files, such as log files, that should be ignored during a comparison.

When the provisioning software performs a Model to Install comparison, the Master Server looks up the list of snapshots for all the components being compared. It sends this list to the Remote Agent on the host where the components are installed. The snapshot information includes rules for any files and directories that should be ignored during the comparison. The Remote Agent then processes these rules. Both Master Server and Remote Agents must be using the same locale for the Model to Install comparison to function properly.

Model to Install Comparison Characteristics

- Compares a snapshot of installed components with the current state of the components
- Allows the flexibility of comparing either an individual component or any set of components installed on a host
- Compares at a high level (component and directory lists) or a low level (file contents)
- Processes comparison on the Remote Agent
- Requires that components created snapshots when installed on a Remote Agent.

You will want to use a Model to Install comparison in the following situation.

- If a configuration was working as expected and is no longer working, you might want to determine if there have been out-of-band changes.

Component Types and Model to Install Comparisons

A Model to Install comparison compares a component model to a specific installation of a component. This section describes how a component type effects a Model to Install comparison.

During a snapshot, the current state of the metabase is exported into an XML file. During a Model to Install comparison, the metabase is re-exported and compared against the original XML file.

IIS Types Introduction

During a snapshot, the current state of the metabase is exported into an XML file. During a Model to Install comparison, the metabase is re-exported and compared against the original XML file. The standard XML differentiator is used to generate differences between these files.

COM+ Application

The COM+ Application is re-exported on the target system as an MSI file and compared against the MSI file used to install the application. A Model to Install comparison will only indicate that there were differences (that is, the two binary files are different). It will not indicate the details of the differences.

COM Component

The COM component is compared as a binary file against the file used during installation. A Model to Install comparison will only indicate that there were differences (that is, the two binary files are different). It will not indicate the details of the differences.

H Keys

During a snapshot, the current state of registry key (and its children) is exported into an XML file. During a Model to Install comparison, the registry key is re-exported and compared against the original XML file. The standard XML differentiator is used to generate differences between these files.

Registry Settings

Model to Install comparisons are not

	supported for * .reg files. Snapshots are not taken during installation. Without a snapshot, there is nothing to compare in a Model to Install comparison. If you need to compare registry changes, use the built-in Registry keys type.
MSI Files or Windows Installer Files	Model to Install comparisons are not supported for Windows Installer files. Snapshots are not taken during installation. Without a snapshot, there is nothing to compare in a Model to Install comparison. Since the provisioning system does not have first-hand knowledge of the actions taken during the installation, it cannot determine what needs to be captured.
Windows Batch Files	Model to Install comparisons are not supported for Windows Batch files. Snapshots are not taken during installation. Without a snapshot, there is nothing to compare in a Model to Install comparison.
Windows Scripting Host Files	Model to Install comparisons are not supported for WSH files. Snapshots are not taken during installation. Without a snapshot, there is nothing to compare in a Model to Install comparison.
WebLogic WAR Files	Model to Install comparisons with WebLogic WAR Files use the standard file and directory approach.
WebLogic Web Application Settings	The relevant settings for the application will be exported into a file, which will be compared to the file that contained the settings during deployment.
WebLogic JAR Files	Model to Install comparisons with WebLogic JAR Files use the standard file and directory approach.
WebLogic EJB Settings	The relevant settings for the EJB will be exported into a file, which will be compared to the file that contained the settings during deployment.
WebLogic EAR Files	Model to Install comparisons with WebLogic EAR Files use the standard file and directory approach.

WebLogic Enterprise Application Settings The relevant settings for the Application will be exported into a file, which will be compared to the file that contained the settings during deployment.

▼ How to Create a Model to Install Comparison

This procedure describes creating a comparison by using the browser interface. You can also create comparisons by using the following command.

`cmp . ds . add` – Creates a new comparison.

For a detailed description of this command, see Chapter 5, “cmp: CLI Commands for Running Comparisons,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin The Remote Agents from each host must be set to use the same locale for Install to Install comparisons to function properly.

To create a new comparison, you must belong to a user group that has Create, Edit, Delete permission on comparisons.

- Steps**
- 1. In the navigation menu, click Comparisons.**
The Comparisons page is displayed.
 - 2. Type the name and description of the comparison you want to create and click Create.**
The comparison’s Details page is displayed.
 - 3. In the Style of Comparison area, click Model to Install.**
 - 4. Select the level of detail that the comparison will use.**
 - **Directory and File properties.** Confirms that both hosts have the same files and directories and that the files and directories are the same sizes.
 - **File Contents.** Compares the contents of files on both hosts.

Note – If the component you are using in a comparison includes its own rules about files to ignore, those rules are used in addition to any ignore rules you define for this comparison.

- 5. Select the scope of the comparison.**

- **Host Set.** Compares the last known state of the specified host set to its current state.
 - a. Select Host Set from the Scope menu.
 - b. Select the host set that you want to compare.
 - **Host.** Compares the last known state of the specified host to its current state.
 - a. Select Host from the Scope menu.
 - b. Type the name of the host that you want to compare.
 - **Component.** Compares the last known state of the specified component to its current state.
 - a. Select Component from the Scope menu.
 - b. Type the name of the host that contains the component and click Apply Change.
 - c. Select the component from the Component (Path) menu.
6. **(Optional) Type the names of files or directories that you want the comparison to ignore.**
- This feature allows the provisioning system to accommodate changes to things like log files without causing the comparison to fail.
- For more information, see [“Modifying Directories or Files to Ignore During Comparisons”](#) on page 41.
7. **In the Limits area, specify the maximum number of hours or minutes the comparison can run.**
- By default, comparisons are allowed to run for 4 hours.
- If the comparison times out, an error message will display on the Comparisons page.
8. **Choose to save the comparison to run later or run it now.**
- To save the comparison without running it, click Save.
 - To both save the comparison and run it now, click Save & Run.

Install to Install Comparisons

Similar to the Model to Install comparison, the Install to Install comparison allows you to specify comparisons down to the level of the contents of files. Install to Install comparisons compare the files and directories on two of hosts.

The list of elements (files, directories, and so on) from the reference host are grouped into batches on the Master Server and then sent to the Remote Agent of the comparison host for processing. If you have defined some rules about file types or specific directories or files for the provisioning system to ignore, the Master Server will discard the appropriate items before sending the remainder of the list to the comparison host. The comparison results are then sent back to the Master Server.

You can run Install to Install comparisons on different directories on the same Remote Agent to compare the contents of two directories on the same host.

Note – The Remote Agents from each host must be set to use the same locale for Install to Install comparisons to function properly.

Install to Install Comparison Characteristics

- Compares the current state of components on one host with the current state of components on another host
- Allows the flexibility of comparing either an individual component or any set of components installed on a host
- Processes comparison on the comparison host's Remote Agent
- Requires that components created snapshots when installed
- Requires that the Remote Agents use the same locale

You will want to use an Install to Install comparison in the following situations.

- You want to compare a component across two hosts.
For example, you can compare software on a staging host with software on a production host.
- You want to compare two directories that both exist on a single host.

Using Scripts in Install to Install Comparisons

If the components you are comparing include resources that are stored in a package, a database, or an archive such as a ZIP file, you can use scripts to export or transform the resources before the comparison occurs. Use the `execNative` step to direct the provisioning software to perform native commands such as unzipping or exporting file contents.

You can run any native command within the `prepare` and `cleanup` blocks. A common usage of these blocks before and after a comparison is to extract the files from a compressed archive and output them to a specified output file, then delete that file during the cleanup.

Note – Unlike the `execNative` contained in a plan or component, this `execNative` instance must be contained within either a `prepare` or a `cleanup` block, and cannot make use of any variable substitutions.

When you are performing an Install to Install comparison using the browser interface, you can select a checkbox labeled `Include Temporary Files Generated By Scripts During Comparison`. When you select this option, the page refreshes and provides two fields into which you can enter the `prepare` and `cleanup` blocks.

You can also specify alternate `prepare` and `cleanup` blocks for the comparison host by deselecting the appropriate boxes.

For more information about `execNative`, see Chapter 2, “Shared Schema Used by Components and Simple Plans,” in *Sun N1 Service Provisioning System 5.1 XML Schema Reference Guide*.

EXAMPLE 2-1 A `prepare` Block for an Install to Install Comparison

This `prepare` block specifies the output file for the result of the `execNative`. The `execNative` lists the contents of the `/tmp` directory.

```
<prepare>
  <execNative outputFile="file.txt" dir="temp">
    <exec cmd="ls">
      <arg value="-l"></arg>
      <arg value="/tmp"></arg>
    </exec>
  </execNative></prepare>
```

EXAMPLE 2-2 A `cleanup` Block for an Install to Install Comparison

This `cleanup` block then removes the output file, `file.txt`, from the `/tmp` directory.

EXAMPLE 2-2 A cleanup Block for an Install to Install Comparison (Continued)

```
<cleanup>
  <execNative dir="/temp/">
    <exec cmd="rm">
      <arg value="-rf"></arg>
      <arg value="file.txt"></arg>
    </exec>
  </execNative></cleanup>
```

▼ How to Create an Install to Install Comparison

This procedure describes creating a comparison by using the browser interface. You can also create comparisons by using the following command.

`cmp.ds.add` – Creates a new comparison.

For a detailed description of this command, see Chapter 5, “`cmp`: CLI Commands for Running Comparisons,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin The Remote Agents from each host must be set to use the same locale for Install to Install comparisons to function properly.

To create a new comparison, you must belong to a user group with Create, Edit, Delete permission on comparisons.

Steps 1. **In the navigation menu, click Comparisons.**

The HTML user interface displays the comparisons page, which lists the comparisons already checked into the database.

2. **Type the name and description of the comparison you want to create and click Create.**

The comparison’s Details page is displayed.

3. **In the Style of Comparison area, click Install to Install.**

4. **Select the level of detail that the comparison will use.**

- **Directory and File properties.** Confirms that both hosts have the same files and directories and that the files and directories are the same sizes.
- **File Contents.** Compares the contents of files on both hosts.

Note – If the component you are using in a comparison includes its own rules about files to ignore, they will be used in addition to any ignore rules you define for this comparison.

5. **Type the name of the reference host and the path of the directory that you want to compare against the comparison host.**

Ideally, the reference host contains the configuration that you would like to see on the comparison host.

The directory that you type specifies where in the hierarchical, file structure to limit the comparison. If you use the top of the tree structure you will evaluate the entire host.

6. **Type the name of the comparison host and the path to the directory you want to compare against the reference host.**

7. **Select the options available for the comparison.**

- **(Optional) To run a comparison on all the subdirectories within the selected directory, select Include All Subdirectories In Comparison.**
- **(Optional) To resolve symbolic links to determine that what the link points to is also valid, select Follow Symbolic Links Found in Directories.**
- **(Optional) To run a script before or after running this comparison, perform the following actions.**

a. **Click Include Temporary Files Generated by Scripts During Comparison.**

The page will refresh and provide two fields to generate and cleanup scripts.

b. **Enter the scripts into the appropriate fields.**

For information on scripts for prepare and cleanup, refer to [“Using Scripts in Install to Install Comparisons”](#) on page 35.

c. **If you do *not* want the provisioning software to run the same scripts on the comparison host as on the reference host, deselect the appropriate options.**

- If you deselect the Generate checkbox, a field displays for you to provide an alternate generate script for the comparison host.
- If you deselect the Cleanup checkbox, a field displays for you to provide an alternate cleanup script for the comparison host.

If you want to specify resources to ignore during this comparison, see [“Modifying Directories or Files to Ignore During Comparisons”](#) on page 41.

8. (Optional) Type the names of files or directories that you want the comparison to ignore.

This feature allows the provisioning system to accommodate changes to things like log files without causing the comparison to fail.

9. In the Limits area, specify the maximum number of hours or minutes the comparison can run.

By default, comparisons are allowed to run for four hours.

If the comparison times out, an error message will display on the Comparisons page.

10. Click Save.

Managing Comparisons

Running comparisons can often take a significant amount of time. After a comparison has been created, you can run, modify, or view it.

- [“How to Run a Comparison” on page 38](#)
- [“How to View Comparisons” on page 39](#)
- [“How to View the Progress of a Comparison” on page 40](#)
- [“How to Stop a Running Comparison” on page 40](#)
- [“Modifying Directories or Files to Ignore During Comparisons” on page 41](#)

▼ How to Run a Comparison

This procedure describes how to run comparisons by using the browser interface. You can also run comparisons by using the following command.

```
cmp .dj .add – Runs a comparison
```

For a detailed description of this command, see Chapter 5, “cmp: CLI Commands for Running Comparisons,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Note – If you have saved a comparison involving a component, and that component has been upgraded or its version numbers have changed for some other reason, the provisioning software automatically updates the comparison you saved to run against the new version of the component.

Before You Begin You must belong to a user group that has permission to run comparisons on a particular host set.

Steps 1. **From the Comparisons page, select the comparison that you want to run.**

2. **(Optional) Modify the comparison criteria.**

3. **Click Save & Run.**

The comparison's Progress page is displayed.

After the comparison completes, the comparison's Results page is displayed.

Note – Running a comparison does not lock down the ability to perform other operations. You can perform other operations while waiting for a comparison to complete.

4. **(Optional) To view the comparison's results in another format, click either Print or Download.**

5. **Click Done.**

▼ How to View Comparisons

This procedure describes how to view comparisons by using the browser interface. You can also view comparisons by using the following commands.

- `cmp.ds.la` – Displays all comparisons
 - `cmp.ds.lo` – Displays detailed information about a particular comparison
- For a detailed description of these commands, see Chapter 5, “cmp: CLI Commands for Running Comparisons,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Steps 1. **In the navigation menu, click Comparisons.**

The Comparisons page is displayed.

2. **(Optional) To view the details of a specific comparison, click Details in the row listing the plan you're interested in.**

The comparison's Details page is displayed.

▼ How to View the Progress of a Comparison

This procedure describes how to view the progress of a comparison by using the browser interface. You can also view a comparison's history by using the following commands.

- `cmp.dj.la` – Lists running and completed comparisons.
- `cmp.dj.lo` – Displays status of a particular comparison.

For a detailed description of these commands, see Chapter 5, “*cmp: CLI Commands for Running Comparisons*,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

- Steps**
1. **In the navigation menu, expand the Application Deployment category and click Run History.**

All running and completed comparisons are listed. Running comparisons are indicated in the Completed column with text describing the time when the plan started running.

2. **To view the details about a particular plan, click Results.**

▼ How to Stop a Running Comparison

This procedure describes how to run comparisons by using the browser interface. You can also run comparisons by using the following command.

`cmp.dj.del` – Stops a comparison

For a detailed description of this command, see Chapter 5, “*cmp: CLI Commands for Running Comparisons*,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

- Before You Begin**
- To stop a comparison, you must be either the user that started the comparison or a member of the `admin` user group.

- Steps**
1. **From the Run History page, determine the comparison you want to stop.**
 2. **In the Action column, click Progress.**
 3. **Click Cancel Comparison.**

Note – If you cancel a comparison in progress, you are returned to the page listing all the comparisons. Occasionally, it may take a short time for the listing to refresh and show you that the comparison that you canceled has been stopped.

▼ Modifying Directories or Files to Ignore During Comparisons

In addition to describing the level of detail and what hosts or components that you want to compare, you have the option of defining sets of files or directories to exclude from a comparison. You might want to do this if you know that particular resources have many differences that are not important to you. You can save these definitions and reuse them in other comparison definitions.

You can also modify comparisons by using the following command.

```
cmp . ds . mod – Modifies a comparison
```

For a detailed description of this command, see Chapter 5, “cmp: CLI Commands for Running Comparisons,” in *Sun N1 Service Provisioning System 5.1 Command-Line Interface Reference Manual*.

Before You Begin To modify a comparison, you must belong to a user group with Create, Edit, Delete permission on comparisons.

- Steps**
1. **If necessary, select the Model to Install comparison or the Install to Install comparison that you want to modify.**
The comparison’s Details page is displayed.
 2. **Modify the Directories and Files to Ignore During Comparisons table.**
 - **Add a directory or file to the ignore list.**
 - a. **From the comparison’s Details page, type the name of the directory or file you want to ignore and click Add.**
You can also provide an expression to represent the directories or files that you want to ignore by using wildcards (*). For example, if you want to ignore all log files, you can specify *.log to be excluded.

Note – You must specify a complete pattern (with or without wildcards) all the way to the file name. Simply specifying down to the directory level will not exclude the files in that directory unless you add a * to the end. For example, if you want to exclude all the files in /usr/apache/logs, you must specify /usr/apache/logs*.

b. Click Add.

Your specification will be added to the list of directories and files to be ignored.

■ **Remove a directory or file from the ignore list.**

a. Choose which directory or file you want to remove.

Note – You cannot remove or disable a global ignore rule.

b. Click Remove.

3. Click Save.

Troubleshooting Deployment Issues

This appendix describes common error messages and log messages in the N1 Service Provisioning System software.

- “Error Message and Descriptions” on page 43
- “Log Messages and Descriptions” on page 46

Error Message and Descriptions

This component has not been installed on any hosts yet

Description: This error occurs when you attempt to uninstall a component that the Master Server does not recognize as having been installed on the host. The Master Server maintains the installation state of each component in the system. An uninstall of a component cannot be performed if the Master Server database does not contain a record that the component was installed on the host. Similarly, you cannot invoke a control on a component until the Master Server database contains a record that the component was installed on the host.

Solution: To correct the problem, follow the steps below:

1. Verify that the component has been installed on the host.
2. If the component has already been installed on the host, use an empty install block to perform a `markOnly` installation of a component to update the Master Server database. The syntax is as follows:

```
<installSteps name="markOnly" />
```

017007 Plan timed out on hosts

Description: The plan time-out parameter is set for a shorter amount of time than is necessary to run a plan.

Example: For example, a step in a plan has been set to pause for 100 seconds. But, the plan time-out parameter was set to 60 seconds. The plan run times out before the pause has completed.

Solution: Verify that the plan time-out parameter is set to a value that allows the running plan to complete. The default plan time out is 30 minutes. Be sure to change the plan time-out parameter for plans that you expect to run for longer than 30 minutes.

017031 Unable to generate configuration file *filename* for host *hostname*

Description: The configuration template contains a variable that is not declared in the `<varList>` of the component.

Example: For example, you attempt to deploy the `/ant/config.properties` configuration template. However, the configuration template contains a `: [port]` variable that is not declared within the `<varList>` of the `/ant/config.properties` component. The following errors display:

- `: [port]` variable is undefined (040201)
- Configuration file `/ant/config.properties` cannot be generated (017031)
- Plan fails because configuration file cannot be generated (017034)

Solution: To prevent this error, verify that the variables that are used within configuration templates are also declared within the `<varList>` of the component. When checking in a configuration template, the provisioning system automatically scans the file for provisioning system variables and adds them to the `<varList>` of the component.

017074 `java.io.IOException: Not enough space`

Description: The Remote Agent does not have enough physical memory to start the Remote Agent process. The message suggests a disk space problem, but the error occurs because of a physical memory problem.

Example: When running a plan that includes a single `<execNative>` step within the plan, the follow error trace appears in the Master Server log file, `server/bin/cr_server.out`:

```
:NestedTrace: java.io.IOException: Not enough space
  at java.lang.UNIXProcess.forkAndExec(Native Method)
  at java.lang.UNIXProcess.(UNIXProcess.java:54)
  at java.lang.Runtime.execInternal(Native Method)
  at java.lang.Runtime.exec(Runtime.java:546)
```

The following error message is displayed within the browser interface:

```
Problems encountered during plan run or preflight
```

```
The plan (or preflight)
"/system/autogen/ldap_cmd-inst-default-1087486030740" finished with 1
failed host(s). (017034)
```

```
Error executing the exec native command "/usr/bin/pfksh". (017074)
Not enough space
plan : /system/autogen/ldap_cmd-inst-default-1087486030740
version : 1.11 version history | run history
```

Solution: To correct the physical memory problem, reduce the minimum JVM™¹ heap size in the `cr_agent` script. Follow these steps:

1. On the server running the Remote Agent, access the directory that contains the Remote Agent scripts.

```
# cd /N1SPS50-home/agent/bin
```

`N1SPS50-home` is the directory in which you installed the Remote Agent.

2. Modify the JVM heap size property.

```
CR_JVM_MEMORY=-Xms32m -Xmx256m
```

`-Xms 32m` is the minimum heap size setting.

3. Reboot the server.

```
030005 Expecting "0" subplan component builds in component build
array "0".
```

Description: This error occurs when running a plan from the CLI. If you include the `-comp +` and the `-vs +` options when you run a plan with the `cr_cli` command, but the plan does not install any components, this error displays.

Example: For example, the following `cr_cli` plan run command was executed at the prompt:

```
% cr_cli -cmd pe.p.run -u bmclean -p password \
-PID NM:/app/Weblogic/RollingWeblogicRestart \
-tar H:NM:dev-prf30 -pto 600s -nto 600s -comp + -vs +
```

The following error displayed:

```
Expecting "0" subplan component builds in component build array "0". (030005)
```

The same plan, with the same target host, was executed successfully from the provisioning system browser interface. The XML contents of the plan are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated by N1 SPS -->
<executionPlan xmlns="http://www.sun.com/schema/SPS"
name="RollingWeblogicRestart" version="4.1"
description="Restart Weblogic instances on target hosts in series"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" path="/app/Weblogic"
xsi:schemaLocation="http://www.sun.com/schema/SPS plan.xsd">

<simpleSteps executionMode="SERIES">
```

1 The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

```
<call blockName="restartAppServer">
  <installedComponent name="WeblogicUtil" path="/app/Weblogic"/>
</call>
</simpleSteps>
</executionPlan>
```

The plan does not install any components. The plan only invokes a control, or call block, on a previously installed component. Consequently, you do not need to specify components or variable sets.

Solution: To prevent this problem, do not specify the `-comp +` and the `-vs +` options on the command line when running a plan that does not install components.

040201 Substitution variable `:target:port` is undefined

Description: The error occurs when the host attributes of a target host do not resolve the target variables in the component.

Example: For example, you attempt to deploy the `/ant/config.properties` configuration template. The component has a `varList` with a variable that defaults to `: [target:port]`. The host to which we are deploying does not have a host attribute that resolves the `port` variable.

Solution: To prevent this error, verify that the host to which you are deploying has the correct set of host attributes to resolve the target variables in the component.

Log Messages and Descriptions

This section describes logged issues that you might encounter when running deployments.

Memory Issues

`java.lang.OutOfMemoryError`

Description: The Master Server has run out of memory during a deployment. You might find this out-of-memory error in the master server's log file, or you might find a similar error message in the browser interface. This error can be caused by deploying large, complex components or deploying to several hosts at the same time.

Examples of large and complex deployments:

- Deploying a container that has more than 100 referenced components.
- Deploying a `system#directory` component that has a number of large files. Files that are bigger than 250k are considered large files.
- Deploying many plans to many hosts.

An example of this is deploying four or more plans to four or more host sets where each host set contains eight or more hosts and each plan deploys more than 100 resources.

Solution: You can prevent this error in two ways. First, you can perform less complex deployments with fewer hosts and components. Second, you can increase the physical RAM in the Master Server and then increase the `CR_MAX_HEAP_SIZE` value.



Caution – Ensure that your system has enough RAM before increasing the `CR_MAX_HEAP_SIZE` value.

To establish the new `CR_MAX_HEAP_SIZE` value, determine the amount of unused physical RAM that was available when the system ran out of memory. The maximum value for `CR_MAX_HEAP_SIZE` should be 100Mb less than the amount of unused physical RAM.

To set the value, modify the `/NISPS-home/server/bin/cr_server` script by changing the `CR_MAX_HEAP_SIZE` value to the following.

```
CR_MAX_HEAP_SIZE=${CR_MAX_HEAP_SIZE:-new-heap-size}
```


Glossary

abstract component	A component that serves only as a base component for other components to extend. An abstract component cannot be installed and only an abstract component is permitted to declare abstract child elements.
call compatibility	A compatibility type for system service components. This compatibility is also called API compatibility or interface compatibility.
category	A general class in which you can group objects that are stored in multiple folders.
child component	A component that is referenced by a container component. Also called <i>contained component</i> . See also <i>container component</i> .
comparison	A feature that searches for and identifies differences between hosts and component models. The provisioning software supports these three types of comparisons: <ul style="list-style-type: none">■ Model to model – Examines the deployment repository and history that is stored on the master server for two hosts and reports any differences■ Model to install – Compares what the master server reports is installed on a host to what is actually on the host and reports any differences■ Install to install – Examines the contents of two hosts' file systems and reports any differences
component	A logical grouping of source information that defines an application. A component also includes a set of instructions that specifies how to manage the source information. The XML representation of a component includes the following:

	<ul style="list-style-type: none"> ■ List of resources used by the application ■ Installation steps ■ Uninstallation steps ■ Dependencies
component compatibility	A situation where a component can be safely replaced by another. The provisioning software supports two kinds of component compatibility: call compatibility and install compatibility.
component inheritance	The means by which a component obtains attributes and behavior from another component. When you create a component, it inherits any variables, snapshots, and procedures from the associated component type.
component procedure	A program in a component that controls deployment of the component, such as installation, uninstallation, management, and capturing snapshots. Management procedures are defined in the control block.
component repository	A location on the master server where components and their resources are checked in.
component type	A special kind of component that encapsulates behavior that can be reused by other components. A component can inherit the behavior of a component type by extending from it.
component variable	A user-definable name-value pair that is used to make parts of a component accessible and configurable by objects that are external to the component.
composite component	A component that contains only references to other components, both simple and composite. A composite component cannot contain any resources.
composite plan	A plan that is composed solely of subplans, which can be simple or composite subplans. A composite plan is not directly targeted, as each subplan can run on a different set of targets.
configuration generation engine	A software engine on the master server that replaces substitution variable references with the appropriate variable setting values. The engine interacts with the host repository and component repository to resolve values any time that you run a plan to deploy a component.
contained component	A component that is referenced by other components.
container component	A component that contains references to other components.
control	A procedure defined by a component that can be used to manage the deployed applications. For example, a control might be used to start or stop an application. Also called <i>control service</i> .

deployment	Using a plan or component procedure to act on a component. The component's lifecycle includes installation, uninstallation, and application management.
direct-run procedure	A component procedure that can be run directly from the component by using the browser interface.
downstream	In the N1 Service Provisioning System software network hierarchy, the server that is further from the master server. For example, the master server connects downstream to a local distributor. Any remote agents connected to the local distributor are downstream from the local distributor.
execNative call	An optional call out to custom scripts from the XML of a plan or component.
execution plan	See <i>plan</i> .
extend	To base a component on a component type so that the component inherits variables and procedures that are defined by the component type. The component can override variable values and procedure definitions defined by its associated component type.
final component	A component that cannot be extended by another component.
folder	Directory-like containers that enable you to apply permissions to and organize components, plans, and subfolders.
gold server	A reference server that contains files, directories, and other resources that make up an application and that checks in these resources to the master server.
host	A server that is managed by the provisioning software.
host set	A user-defined, logical grouping of hosts that share one or more common attributes, such as physical location or functional group. Use a host set to quickly and easily update applications on all hosts in the set. You can also use a host set to perform model-to-install comparisons between two hosts.
host type	A base class of servers that is bound by a set of common attributes, all of which are user-defined. You can use host types to categorize hosts into logical groupings and to facilitate host searches.
host search	A query run on the host repository that yields a list of hosts whose attributes match those specified by the query. For example, you can use host searches to create a list of hosts that have the same host type, that run the same applications, and that are configured with the same subnet masks.
install compatibility	A compatibility type for component types. This compatibility is also called structural compatibility.

Java Runtime Environment (JRE)	A subset of the Java Development Kit (JDK®) for users and developers who want to redistribute the runtime environment. The Java runtime environment consists of the Java virtual machine (JVM), the Java core classes, and supporting files.
Java Virtual Machine (JVM)	The part of the Java runtime environment (JRE) responsible for interpreting bytecodes.
Jython	An implementation of the high-level, dynamic, object-oriented language, Python, seamlessly integrated with the Java platform. The predecessor to Jython, JPython, is certified as 100% pure Java.
label	A means of marking a component version beyond the provisioning software version number. For example, a component version number describes the version of the component. A label can describe the version of the application that the component represents.
local distributor	<p>The application that is installed on a server. The Local Distributor application acts as a link between other servers in the provisioning software in the following ways:</p> <ul style="list-style-type: none"> ■ master server to remote agents ■ master server to other local distributors ■ local distributor to remote agents <p>Local distributors maximize bandwidth efficiency and speed, and can also provide secure network connections for navigating restricted environments.</p>
master server	The application that is installed on a server that manages the provisioning software. The Master Server application can connect to any of the data center environments managed by the provisioning software. The master server provides centralized data storage, data processing, and user interfaces.
modeling	To create components and plans that represent an application that you want to deploy with the provisioning software.
nested component	A contained component that, when installed, can provide its services only to its container component. A nested contained component defines a finer-grained unit of functionality required by the container component, but is not otherwise useful to other components.
network protocol	A way to transmit data between devices on a network. The N1 Service Provisioning System software uses TCP/IP, SSH, and SSL.
notification email	An email sent by the provisioning software to advise that a system, administrative, or custom event has occurred. The system administrator specifies the rules used to determine when notification emails are sent and the email addresses to which the email is sent.

notification rule	The criteria used by the provisioning software to determine whether an email notification is sent. The system administrator defines the criteria that that is used to determine when an email notification is sent.
parent component	A component that contains references to other components. Also called <i>container component</i> . See also <i>contained component</i> .
physical host	A physical server that is connected to the network. Within the provisioning system, a physical host can act as a remote agent or a local distributor.
plan	A sequence of instructions that is used to manipulate one or more components. A plan can also be a sequence of other plans, which enables common instruction sequences to be shared between one or more plans.
plan executor	The software engine on the master server that runs preflights and deployments.
preflight	The simulated execution of a plan to a simulated UNIX environment that finds and reports any errors or potential errors that might affect the deployment. A preflight always precedes a deployment, but you can run a preflight as a standalone operation.
procedure	See <i>component procedure</i> .
provisioning system	The software applications that, when installed on servers, form the N1 Service Provisioning System software.
remote agent	The application that is installed on any server in the provisioning software to which components are deployed. The Remote Agent application manages tasks, such as installing software, controlling services, and collecting information to deliver to the master server.
resource	A file that is deployed to a host when a plan is executed. The file might be a directory, a symbolic link, or another kind of file.
resource descriptor file	An XML file that specifies the owner, group, and permission settings to use for the files and directories that comprise the resource of a simple component. By using a resource descriptor file, you can override the permissions that are determined at component check-in time.
server	A computer that manages resources and supplies services to a client. In the N1 Service Provisioning System software, a server is a computer on which one of the provisioning software applications has been installed.

session	A period of time that is initiated when you log in. A session persists until you log out or inactivity causes the session to expire. Logically, a session represents the authenticated credentials of a particular user. A session is used to identify the user throughout a series of related requests without reauthentication.
session variable	A variable that is associated with a user session. The user can change session variable values for each login session. Session variable values can also be securely saved for reuse in subsequent sessions.
simple component	A component that contains a single resource. A simple component cannot contain references to other components.
simple plan	A sequential list of steps that are executed on a particular set of target servers. A simple plan does not contain or call other plans.
snapshot	A capture of the resources that are stored on a host during a deployment. The snapshot is used when performing comparisons between a host and its model on the master server (model-to-install).
step	An instruction that can be part of a plan or a component.
substitution variable	A variable that appears in plans, components, or configuration files that is substituted by the configuration generation engine during deployment.
system service	A component that is automatically deployed to all applicable hosts when the hosts are prepared. System services define utility controls and resources that can be used by other components.
targetable component	A component that creates a host that serves as a deployment target for other components when it is installed. When a targetable component is uninstalled, the host it created is automatically deleted.
top-level component	A contained component that, when installed, can be used by any component just as if it had been directly installed by a plan. A top-level contained component defines services that will be used by the container component as well as by other components.
upstream	In the provisioning software network hierarchy, the server that is closer to the master server. For example, the master server is upstream from the local distributor. The local distributor is upstream from any remote agents that are connected to that local distributor.
variable	See <i>component variable</i> .
variable settings	A collection of variable values that can be used to override the default values of one or more component variables. Based on the variable settings that you use, you can specify different values for component variables. You specify the variable settings to use when you run a plan.

virtual host

Services that act as a host for other services. For example, a virtual host can represent an application server that acts as a host for web applications.

XML schema

The language used by the provisioning software to create plans and components.

Index

A

auto-generated plans, *See* generated plans

C

canceling, *See* stopping

cleanup block, 35-36

comparison settings, *See* comparisons

comparisons, 25-26

 creating

 Install to Install, 36-38

 Model to Install, 32-33

 Model to Model, 27-28

 ignoring directories or files, 41-42

 Install to Install, 34-38

 Model to Install, 28-33

 Model to Model, 27-28

 running, 38-39

 stopping, 40-41

 synchronizing hosts, 26

 viewing, 39

component procedures, running, 12

components, deploying, 11-16

creating

 comparisons

 Install to Install, 36-38

 Model to Install, 32-33

 Model to Model, 27-28

 generated plans, 15-16

D

deleting, plans, 20-21

deploying, components, 11-16

difference settings, *See* comparisons

diffs, *See* comparisons

direct-run procedures, using, 12

E

error messages, descriptions, 43-46

excluding, *See* ignoring

executing

See running

G

generated plans

 creating, 15-16

 running, 15

I

ignoring files and directories,

 comparisons, 41-42

Install to Install comparisons, 34-38

 creating, 36-38

 using scripts, 35-36

 prepare and cleanup, 35-36

L

log messages, 46-47

M

memory issues, troubleshooting, 46-47

Model to Install comparison, 28-33

 snapshots, 28-33

Model to Install comparisons, creating, 32-33

Model to Model comparisons, 27-28

 creating, 27-28

P

plans

 creating

 generated, 15-16

 deleting, 20-21

 running, 17-19

prepare block, 35-36

problems, *See* error messages

R

running

 comparisons, 38-39

 component procedures, 12

 generated plans, 15, 17-19

 plans, 17-19

S

snapshots, 28-33

stopping, comparisons, 40-41

synchronizing hosts, comparisons, 26

T

troubleshooting

 error messages, 43-46

 log messages, 46-47

 memory issues, 46-47

V

viewing, comparisons, 39

W

workarounds, 43-46