Sun N1 Service Provisioning System User's Guide and Release Notes for the Solaris Plug-In 3.0

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

Part No: 819–2900–10
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, N1, Java, 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 Etats-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, N1, Java, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-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 Etats-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 Overview of Solaris Plug-In 11
Purpose of the Solaris Plug-In 11
What the Solaris Plug-In Includes 11
Solaris Patches 12
Solaris Packages 12
Solaris 10 Zones 12
Requirements for Using the Solaris Plug-In 13
Requirements for Master Server 13
Requirements for Target Hosts 13

2 Release Notes 15
Installation Issues 15
Runtime Issues 15

3 Installing and Configuring the Solaris Plug-In 17
Acquiring the Solaris Plug-In 17
Adding the Solaris Plug-In to the N1 Service Provisioning System 18
▼ How to Import the Solaris Plug-In Using the Browser Interface 18
▼ How to Import the Solaris Plug-In Using the CLI 18
Upgrading Considerations 19
Upgrading Existing Local Zones 19
▼ How to Upgrade Global Zones 19
▼ How to Upgrade a Local Zone that is Running 19

▼ How to Upgrade a Local Zone that is Not Running 20
Accessing Previous Versions of a Component 21
▼ How to Access Previous Versions of a Component 21
Customizing the Solution for Your Environment 22
▼ How to Customize the `config.properties` File 22

4 Using the Solaris Plug-In to Deploy Patches 23
Introduction to Solaris Patches 23
Deploying Solaris Patches 23
▼ How to Deploy a Solaris Patch from the Browser Interface 24
Solaris Patch Component Type 24
Component Variables 25
Component Procedures 25

5 Using the Solaris Plug-In to Deploy Packages 27
Introduction to Solaris Packages 27
Deploying Solaris Packages 28
▼ How to Deploy a Solaris Package or Datastream Package Without a Response File 28
▼ How to Deploy a Solaris Package or DataStream Packages With a Response File 29
Solaris Package Components 31
Package Component and Datastream Package Component 31
Response File Component 33
Solaris Package Container Components 34
Package Container Component and Datastream Package Container Component 34

6 Using the Solaris Plug-In to Deploy Zones 37
Introduction to Zones Partitioning 37
Creating and Managing Solaris Zones 38
▼ How to Define a Host as a Global Zone 38
▼ How to Create a Solaris Local Zone 38
Using the CLI to Work With Solaris Zones 39
Solaris Zone Component Types 40
Solaris Zone Container Component 41
Component Procedures 41
Solaris Zone Plans 42

Hosts and Solaris Zones 42
Troubleshooting Solaris Zones 43

Index 45

Preface

The *Sun N1 Service Provisioning System User's Guide and Release Notes for the Solaris Plug-In 3.0* contains information about installing, configuring, and using the N1™ Service Provisioning System to provision Solaris™ patches, packages, and Solaris 10 zones.

Who Should Use This Book

The main audience for the *Sun N1 Service Provisioning System User's Guide and Release Notes for the Solaris Plug-In 3.0* includes system administrators and operators of N1 Service Provisioning System software who want to use the provisioning system to deploy Solaris patches and packages and to create Solaris 10 zones. These users are expected to be familiar with the following:

- The N1 Service Provisioning System product
- Standard UNIX® and Microsoft Windows commands and utilities
- General concepts associated with Solaris patches
- General concepts associated with Solaris packages
- General concepts associated with Solaris 10 zones

Before You Read This Book

If you are not already familiar with using the N1 Service Provisioning System software, read the following books:

- *Sun N1 Service Provisioning System 5.1 System Administration Guide*

- *Sun N1 Service Provisioning System 5.1 Operation and Provisioning Guide*
- *Sun N1 Service Provisioning System 5.1 Release Notes*

How This Book Is Organized

Chapter 1 provides an overview of the Solaris Plug-In and its contents and includes system requirements for using the Solaris Plug-In.

Chapter 2 describes installation and runtime issues.

Chapter 3 explains how to install and configure the Solaris Plug-In.

Chapter 4 describes how to use the components that are provided with the Solaris Plug-In to deploy patches.

Chapter 5 describes how to use the components and containers provided with the Solaris Plug-In to deploy packages.

Chapter 6 describes how to use the components and plans that are provided with the Solaris Plug-In to deploy zones.

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/supportraining/`	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–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `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*. Perform a *patch analysis*. Do *not* 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–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`

TABLE P–2 Shell Prompts *(Continued)*

Shell	Prompt
Bourne shell and Korn shell superuser prompt	#

CHAPTER 1

Overview of Solaris Plug-In

This chapter explains general information about using N1 Service Provisioning System to provision Solaris patches, packages, and zones. The chapter includes the following information:

- "Purpose of the Solaris Plug-In" on page 11
- "What the Solaris Plug-In Includes" on page 11
- "Requirements for Using the Solaris Plug-In" on page 13

Purpose of the Solaris Plug-In

The Solaris plug-in adds components, plans, and system services to the N1 Service Provisioning System that enable you to use the provisioning system to complete the following tasks:

- Deploy Solaris patches
- Deploy Solaris packages
- Create and manage Solaris 10 zones

What the Solaris Plug-In Includes

The Solaris Plug-In creates the `/com/sun/solaris` directory.

For details about using the plans, components, and host set, host types, and host searches listed in the following sections see the following chapters:

- Chapter 4

- Chapter 5
- Chapter 6

Solaris Patches

`com.sun.solaris#Patch` component type
Component type to deploy a Solaris patch.

Solaris Packages

`com.sun.solaris#Package` component type
Component type to deploy a Solaris package.

`com.sun.solaris#StreamPackage` component type
Component type to deploy a Solaris datastream package.

`com.sun.solaris#ResponseFile` component type
Component type to create a component that deploys a file that includes the responses that are necessary to install a package without user interaction.

`com.sun.solaris#PackageWithResp` container
Container to deploy a `com.sun.solaris#Package` component with its corresponding `com.sun.solaris#ResponseFile` component.

`com.sun.solaris#StreamPackageWithResp` container
Container to deploy a `com.sun.solaris#StreamPackage` component with its corresponding `com.sun.solaris#ResponseFile` component.

Solaris 10 Zones

`Container` component
Main component used to create and manage zones. For more information, see "Solaris Zone Container Component" on page 41.

`container_util` component
Component that contains utility scripts for zones

`Container-create` plan
Plan that creates zones

`Container-attach` plan
Plan that attaches zones

`Container-delete` plan
Plan that deletes zones

`Container-detach` plan
Plan that detaches zones

`Container-activate` plan
Plan that activates zones

`Container-deactivate` plan
Plan that deactivates zones

`com.sun.solaris#global_zones` host set
Host set for global zones

`com.sun.solaris#global_zones` host search
Host search for global zones

`com.sun.solaris#global_zone` host type
Host type for global zones

`com.sun.solaris#local_zone` host type
Host type for local zones

Requirements for Using the Solaris Plug-In

Requirements for Master Server



Caution – The Solaris Plug-In 3.0 runs in the Sun N1 Service Provisioning System 5.1 environment. Do not install the Solaris Plug-In 3.0 on any other version of the N1 Service Provisioning System.

To deploy Solaris patches, packages, and zones, make sure that the following two parts of the N1 Service Provisioning System are installed on the Master Server from which you want to deploy:

- CLI (command-line interface)
- Remote Agent

Requirements for Target Hosts

Any host on which you intend to deploy Solaris patches and packages must meet the following requirements:

- Run the Solaris 7, Solaris 8, Solaris 9, or Solaris 10 Operating System (OS).

Note – Solaris 7 and Solaris 8 on SPARC® based servers only

- Run the Remote Agent. The Remote Agent must be run as the root user.

Any host on which you intend to deploy Solaris 10 zones must meet the following requirements:

- Run the Solaris 10 Operating System.
- Meet the operating system requirements for hosting zones.
- Run the Remote Agent. If you are using TCP/IP as the network protocol for the provisioning system, run the Remote Agent as the root user. If you do not want to run the Remote Agent as the root user, use SSH as the network protocol.

CHAPTER 2

Release Notes

This chapter describes late-breaking news and known issues with the Solaris Plug-In.

Installation Issues

There are no known installation issues.

Runtime Issues

There are no known runtime issues.

CHAPTER 3

Installing and Configuring the Solaris Plug-In

This chapter contains the following information:

- "Acquiring the Solaris Plug-In" on page 17
- "Adding the Solaris Plug-In to the N1 Service Provisioning System" on page 18
- "Upgrading Considerations" on page 19
- "Customizing the Solution for Your Environment" on page 22

Acquiring the Solaris Plug-In

The Solaris Plug-In is packaged as a *plug-in* to the N1 Service Provisioning System software. Plug-ins are packaged in Java™ Archive (JAR) files. The plug-in files for the Solaris Plug-In are available from the Sun N1 Service Provisioning System 5.1 Supplement CD or from the Sun Download Center.

The Solaris Plug-In is available for import from two different JAR files. Choose the correct file depending upon your situation.

- If you are importing the Solaris Plug-In for the first time, acquire the `com.sun.solaris_3.0.jar` file.
- If you have already imported the previous version of the Solaris Plug-In, acquire the `com.sun.solaris_2.0_3.0.jar` file.

Adding the Solaris Plug-In to the N1 Service Provisioning System

To make a given plug-in known to the N1 Service Provisioning System, you need to import the plug-in to the Master Server. If you have already imported a previous version of the Solaris Plug-In, you will need to upgrade to the new plug-in.

▼ How to Import the Solaris Plug-In Using the Browser Interface

To import or upgrade a plug-in, follow these steps as explained in detail in Chapter 5, "Plug-In Administration," in *Sun N1 Service Provisioning System 5.1 System Administration Guide*.



Caution – The Solaris Plug-In 3.0 runs in the Sun N1 Service Provisioning System 5.1 environment. Do not install the Solaris Plug-In 3.0 on any other version of the N1 Service Provisioning System.

Steps

1. **In the Administrative section of the browser interface main window, click Plug-ins.**
2. **In the Action column of the Plug-ins page, click Import.**
3. **Browse to the location where you downloaded the JAR file.**
   - If you are importing the Solaris Plug-In for the first time, select the `com.sun.solaris_3.0.jar` file.
   - If you have already imported a previous version of the Solaris Plug-In, select the `com.sun.solaris_2.0_3.0.jar` file.
4. **Click the Continue to Import button.**
   When the import completes successfully, a plug-in details page appears and shows you the objects that the plug-in provides.

▼ How to Import the Solaris Plug-In Using the CLI

You can also import a plug-in archive file by using the command line.

Step

- **To import a plug-in file from the CLI, type:**

  `% cr_cli -cmd plg.p.add -path` *plugin-filename* `-u` *username* `-p` *password*

- If you are importing the Solaris Plug-In for the first time, *plugin-filename* is `com.sun.solaris_3.0.jar`.
- If you have already imported the previous version of the Solaris Plug-In, *plugin-filename* is `com.sun.solaris_2.0_3.0.jar`.

Upgrading Considerations

Upgrading Existing Local Zones

If you have created global and local zones by using the Solaris Plug-In in the Sun N1 Service Provisioning System 5.0, after you migrate the Master Server to the Sun N1 Service Provisioning System 5.1 and you import the Solaris Plug-In 3.0, you must upgrade the zones. You can then control the upgraded zones with the Solaris Plug-In 3.0.

▼ How to Upgrade Global Zones

Steps

1. **In the Common Tasks section of the browser interface main window, click the Sun Solaris icon.**
2. **Click the Solaris Containers: Create and Manage link.**
3. **In the Reference Components list, click `/com/sun/solaris/container_util`.**
4. **In the Component Procedures list, click the Run action next to install:default.**
5. **On the Run Details page select the global zone target hosts to upgrade.**
6. **Click the Run Plan (includes preflight) button.**

▼ How to Upgrade a Local Zone that is Running

Use the following steps to upgrade a local zone that is running.

Before You Begin

After migrating to the Sun N1 Service Provisioning System 5.1, upgrade the Remote Agents. For instructions see "Upgrading Remote Agents and Local Distributors" in *Sun N1 Service Provisioning System 5.1 Installation Guide*.

Steps

1. **In the Common Tasks section of the browser interface main window, click the Sun Solaris icon.**
2. **Click the Solaris Containers: Create and Manage link.**
3. **In the Component Procedures list, click the Run action next to markOnly: install.**
4. **Select the target global zone host in which the local zone that you want to upgrade is installed.**
5. **Select the local zone host name from the Plan Variables settings.**

   If desired local zone names are not present, you can add the zone names. To add the zone names, follow these steps:

   a. **Click Select From List in the plan variable settings table.**
   b. **In the Variable Setting window, click Create Set.**
   c. **Type the zone set name in the New Set Name field.**
   d. **Click the box next to Zone Name prompt and type the zone name.**

   Note – Zone names become network Zone Host Names during local zone creation. Zone Host Names should resolve to matching available local network IP addresses.

   ---

6. **Click the Run Plan (includes preflight) button.**
7. **In the Common Tasks section of the browser interface main window, click the Sun Solaris icon.**
8. **Click the Solaris Containers: Create and Manage link.**
9. **In the Component Procedures list, click the Run action next to activate.**
10. **On the Plan Details page, in the Current Installations list, select the local zones to activate.**
11. **Click the Run Selected Installations button.**

▼ How to Upgrade a Local Zone that is Not Running

Use the following steps to upgrade a local zone that has been deactivated.

Steps

1. **In the Common Tasks section of the browser interface main window, click the Sun Solaris icon.**

2. **Click the Solaris Containers: Create and Manage link.**
3. **In the Component Procedures list, click the Run action next to attach: install.**
4. **Select the target global zone host in which the local zone that you want to upgrade is installed.**
5. **Select the local zone host name from the Plan Variables settings.**

   If desired local zone names are not present, you can add the zone names. To add the zone names, follow these steps:

   a. **Click Select From List in the plan variable settings table.**
   b. **In the Variable Setting window, click Create Set.**
   c. **Type the zone set name in the New Set Name field.**
   d. **Click the box next to Zone Name prompt and type the zone name.**

   Note – Zone names become network Zone Host Names during local zone creation. Zone Host Names should resolve to matching available local network IP addresses.

   ---

6. **Click the Run Plan (includes preflight) button.**

   The provisioning system upgrades the local zone from the 2.0 version to the 3.0 version.

   Note – If you want to activate the local zone, run the activate Component Procedure.

Accessing Previous Versions of a Component

When you upgrade to a new version of a plug-in, the Common Tasks page for the plug-in is updated to provide links to components that are installed with the new plug-in version. This feature enables you to easily view components that contain the most current features and improvements. Components that rely on old functionality are not linked to from the Common Tasks page.

▼ How to Access Previous Versions of a Component

If you have trouble finding a component that you previously installed, the component was likely created from the previous version of the plug-in. Follow these steps to view or work with a previous version of the component:

Steps

1. **Click the appropriate component procedure on the Common Steps Tasks page.**
   The Component Detail page appears.
2. **On the Component Details page, click Version History.**
   A list of components and versions appears.
3. **Click the appropriate link.**
   - To perform a task with a component, click the version number or the details link that applies to the component that you want to use.
   - To view where a component is installed, click Where Installed.

Customizing the Solution for Your Environment

Before you deploy Solaris zones from the Master Server, you must modify the Master Server `config.properties` file.

▼ How to Customize the `config.properties` File

Steps

1. **On the Master Server, using a text editor, access the `config.properties` file.**
   By default, the configuration file for the Master Server resides in the following directory:
   `/opt/SUNWn1sps/N1_Grid_Service_Provisioning_System_5.0/server/config`
2. **Add the following lines to the `config.properties` file:**

   ```
   config.allowSessionIDOnHosts=masterserver
            pe.defaultPlanTimeout=6000
   ```

   The `config.allowSessionIDOnHosts` variable enables the plug-in to communicate with the Master Server through the N1 Service Provisioning System Remote Agent and Command Line Interface (CLI).

   By default, the timeout value for plans is set to 1800 seconds or 30 minutes. In some instances, such as when executing a plan on older hardware, the amount of time that is needed for a plan to complete exceeds 30 minutes. To increase the plan timeout value, add the `pe.defaultPlanTimeout` line to the `config.properties` file. In the example that is shown, the value is set to 100 minutes.
3. **After you modify the `config.properties` file, you must restart the Master Server.**

CHAPTER 4

Using the Solaris Plug-In to Deploy Patches

This chapter explains how to use the Solaris Plug-In within the N1 Service Provisioning System to deploy a patch. The chapter contains the following information:

- "Introduction to Solaris Patches" on page 23
- "Deploying Solaris Patches" on page 23
- "Solaris Patch Component Type" on page 24

Introduction to Solaris Patches

A Solaris patch is a collection of files and directories that replaces or updates existing files and directories that are preventing proper execution of the Solaris OS or other software. The Solaris Plug-In enables you to use the N1 Service Provisioning System to install a Solaris patch on multiple servers. For more information about Solaris patches and how to acquire them, see Chapter 20, "Managing Solaris Patches (Overview)," in *System Administration Guide: Basic Administration*.

Deploying Solaris Patches

The N1 Service Provisioning System uses the `patchadd`(1M) command to install patches and the `patchrm`(1M) command to uninstall patches.

▼ How to Deploy a Solaris Patch from the Browser Interface

Before You Begin

Obtain the patch that you want to deploy.



Caution – The N1 Service Provisioning System does not check for patch dependencies. Be sure to install patches in the proper sequence to account for patch dependencies.

Steps

1. **In the left control panel, under Common Tasks, click the Sun Solaris icon.**
   The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
2. **Click the Solaris Patch: Create Patch Component link.**
3. **Define the component.**
4. **In the left control panel, under Application Deployment, click Components.**
   A list of components appears.
5. **Click the component that you just created.**
6. **(Optional) Click the Edit button to change any of the component variables.**
   For details about the component variables, see "Component Variables" on page 25.
7. **Choose the Component Procedure that you want to run and click the Run link.**
   For details about the component procedures, see "Component Procedures" on page 25.
8. **Complete the plan run page to run the procedure.**

Solaris Patch Component Type

The `com.sun.solaris#Patch` component type is the only component type delivered by the Solaris plug-in for use with Solaris patches. To deploy Solaris patches using the N1 Service Provisioning System, create components of type `com.sun.solaris#Patch`. For more information about creating components, see Chapter 4, "Plans," in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide*.

Component Variables

`installName`	The name to use for the patch when it is installed. The default value is the name of the patch component.
`installPath`	The path on the target host where the patch will be downloaded.
`rootDirectory`	The path on the target host where the patch will be installed.
`validateFiles`	This variable enables you to supply the `-u` option to `patchadd`(1M) when the patch is added to the target host. The `-u` option turns off file validation.`patchadd` applies the patch even if some of the files to be patched have been modified since their original installation. Valid values are `true` and `false`.
`patchID`	This value is the ID of the patch.

Note – Changing the value of this variable might cause the patch deployment to fail.

Component Procedures

`default: install`	Use this procedure to install a patch on a target system or systems.
`markOnly: install`	If you manually installed a patch on a target system or systems but want the provisioning system database to be aware that the patch is installed on those systems, use this procedure.
`default: uninstall`	Use this procedure to uninstall a patch from a target system or systems.
`markOnly: uninstall`	If you manually uninstalled a patch from a target system or systems but want the provisioning system database to be aware that the patch has been uninstalled from those systems, use this procedure.

CHAPTER 5

Using the Solaris Plug-In to Deploy Packages

This chapter explains how to use the Solaris package specific components, containers, and system services that are provided with the Solaris Plug-In. The chapter contains the following information:

- "Introduction to Solaris Packages" on page 27
- "Deploying Solaris Packages" on page 28
- "Solaris Package Components" on page 31
- "Solaris Package Container Components" on page 34

Introduction to Solaris Packages

Sun and its third-party vendors deliver software products in a form that is called a *package*. The term packaging generically refers to the method for distributing and installing software products to systems where the products will be used. A package is a collection of files and directories in a defined format. This format conforms to the application binary interface (ABI), which is a supplement to the System V Interface Definition. The Solaris OS provides a set of utilities that interpret this format and provide the means to install a package, to remove a package, or to verify a package installation.

Solaris packages are created in a file system format. The package is a directory or series of directories that contain the files and scripts necessary to install the package. You can convert the file system into a single datastream by using the `pkgtrans`(1) command.

Some Solaris packages contain a `request` script that asks you questions prior to installation. If you want to use the N1 Service Provisioning System to install a package that has a `request` script, you must create a response file that contains the responses to the questions asked by the `request` script. Use the `pkgask`(1M) command to create the response file. For more information about creating and using response files, see "Avoiding User Interaction When Adding Packages (pkgadd)" in *System Administration Guide: Basic Administration*.

Deploying Solaris Packages

The N1 Service Provisioning System uses the `pkgadd`(1M) command to install Solaris packages and the `pkgrm`(1M) command to remove Solaris packages.

Some Solaris packages contain a `request` script that asks you questions prior to installation. If you want to use the N1 Service Provisioning System to install a package that has a `request` script, you must create a response file that contains the responses to the questions asked by the `request` script. Use one of the following procedures to install a Solaris package:

- "How to Deploy a Solaris Package or Datastream Package Without a Response File" on page 28
- "How to Deploy a Solaris Package or DataStream Packages With a Response File" on page 29

▼ How to Deploy a Solaris Package or Datastream Package Without a Response File

If the package you want to install does not include a `request` script that requires responses, follow the steps below to use the N1 Service Provisioning System to install the package.

Before You Begin

Obtain the package that you want to install.



Caution – The N1 Service Provisioning System does not check for package dependencies. Be sure to install packages in the proper sequence to account for package dependencies.

Steps

1. **In the left control panel, under Common Tasks, click the Sun Solaris icon.**

   The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.

2. **Create a Package component.**
   - If you want to deploy a Solaris file system package, click the Solaris Package: Create Package Component link.
   - If you want to deploy a Solaris datastream package, click the Solaris Datastream Package: Create Datastream Package Component link.
3. **In the left control panel, under Application Deployment, click Components.**
4. **From the list of components, click the component that you created.**
5. **(Optional) Edit the component variables by clicking the Edit button.**

Note – If you are deploying a datastream package, you might need to edit the `pkgName` component variable for the package to deploy successfully.

For more information, see "Component Variables" on page 31.

6. **In the Component Procedures section, click the Run link associated with the component procedure that you want to run.**
   For more information, see "Component Procedures" on page 33.

▼ How to Deploy a Solaris Package or DataStream Packages With a Response File

If the package that you want to install includes a `request` script that asks you questions prior to installation, follow the steps below to install the package.

Before You Begin

Obtain the package that you want to install.

Use the `pkgask`(1M) command to create the response file. For more information about creating and using response files, see "Avoiding User Interaction When Adding Packages (pkgadd)" in *System Administration Guide: Basic Administration*



Caution – The N1 Service Provisioning System does not check for package dependencies. Be sure to install packages in the proper sequence to account for package dependencies.

Steps

1. **In the left control panel, under Common Tasks, click the Sun Solaris icon.**
   The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.

2. **Create a Package component.**
   - If you want to deploy a Solaris file system package, click the Solaris Package: Create Package Component link.
   - If you want to deploy a Solaris datastream package, click the Solaris Datastream Package: Create Datastream Package Component link.
3. **In the left control panel, under Common Tasks, click the Sun Solaris icon.**

   The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
4. **Click the Response File: Create Response File Component link.**

   Define the response file component to reference the response file that you created for use with the package you want to install.
5. **In the left control panel, under Common Tasks, click the Sun Solaris icon.**

   The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
6. **Create a Container that includes the Package component and the Response File component.**
   - If you want to deploy a Solaris file system package, click the Package with Response File: Create Package With Response File Container link.
   - If you want to deploy a Solaris datastream package, click the Datastream Package with Response File: Create Datastream Package With Response File Container link.
7. **In the left control panel, under Application Deployment, click Components.**
8. **From the list of components, click the container that you created.**
9. **(Optional) Edit the component variables by clicking the Edit button.**

   Note – If you are deploying a datastream package, you might need to edit the `pkgName` component variable for the package to deploy successfully.

   For more information, see "Component Variables" on page 34.
10. **In the Component Procedures section, click the Run link associated with the Component Procedure that you want to run.**

    For more information, see "Component Procedures" on page 34.

Solaris Package Components

Package Component and Datastream Package Component

Use the Package components to create components that contain the packages that you want to install or remove. For more information about creating components, see Chapter 4, "Plans," in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide*.

The Solaris Package: Create Package Component link creates a component of type `com.sun.solaris#Package`. This component references a file system package. A file system package contains a directory or series of directories that contain the files and scripts necessary to install the package.

The Solaris Datastream Package: Create Datastream Package Component link creates a component of type `com.sun.solaris#StreamPackage`. This component references a datastream package. A datastream package is a package that has been converted from a file system package into one data stream. For more information about datastream packages, see the `pkgtrans`(1) man page.

The Package components and the Datastream Package components contain the same component variables and component procedures. The variables and procedures are described below.

Component Variables

`installName`	The name to use for the package when it is installed. The default value is the name of the package component.
`installPath`	The path on the target host where the package will be downloaded.
`installRoot`	The path on the target host where the package will be installed.
`pkgName`	The name of the package. When installing a datastream package, you might need to change the value of this variable. When installing a datastream package using the `pkgadd` command, you are prompted to enter the package name. The provisioning system cannot prompt you during the package installation for the package name, so you must supply the actual package name here.

You can find the package name for the datastream package in the header of the datastream file. In the following example, the package name is `SMCcvs`.

```
% head cvs-1.11.17-sol9-sparc-local
# PaCkAgE DaTaStReAm
SMCcvs 1 16852
# end of header
NAME=cvs
ARCH=sparc
VERSION=1.11.17
CATEGORY=application
VENDOR=Joe Smith et al
EMAIL=joe@smith.work.net
PSTAMP=Bob Miller
```

`verboseMode` This variable enables you to supply the `-v` option to `pkgadd`(1M) when the package is added to the target host. The `-v` option traces all of the scripts that `pkgadd` executes, located in the `pkginst/install` directory. Use this option for debugging the procedural and non-procedural scripts. Valid values are `true` and `false`.

The remaining component variables correspond directly to the variable values that the provisioning system sets in an administration file that will be used to install the package. The provisioning system creates an administration file to provide information to the `pkgadd` command about how the installation should proceed. For more information about administration files, see "Using an Administration File" in *System Administration Guide: Basic Administration*. For descriptions of the following variables and valid values for each variable, see the `admin`(4) man page.

```
mail
instance
partial
runlevel
independ
redepend
space
setuid
conflict
action
basedir
```

Note – The values set by the provisioning system for the administration file variables do not match the default values that are set in the administration file that is installed with the Solaris OS in the `/var/sadm/install/admin` directory. The values set by the provisioning system enable the package installation to complete without any interaction. For more information about the default values that are set in the administration file that is installed with the Solaris OS, see the `admin`(4) man page.

Component Procedures

`default: install`	Use this procedure to install a package on a target system or systems.
`mark Only: install`	If you manually installed a package on a target system or systems but want the provisioning system database to be aware that the package is installed on those systems, use this procedure.
`default: uninstall`	Use this procedure to uninstall a package from a target system or systems.
`markOnly: uninstall`	If you manually uninstalled a package from a target system or systems but want the provisioning system database to be aware that the package has been uninstalled from those systems, use this procedure.

Response File Component

You create a Response File component when you have created a response file to use with a package. Some Solaris packages contain a `request` script that asks you questions prior to installation. If you want to use the N1 Service Provisioning System to install a package that has a `request` script, you must create a response file that contains the responses to the questions asked by the `request` script. Use the `pkgask`(1M) command to create the response file that you will reference in the Response File Component. For more information about response files, see "Avoiding User Interaction When Adding Packages (pkgadd)" in *System Administration Guide: Basic Administration*.

The Response File: Create Response File Component creates a component of the type `com.sun.solaris#ResponseFile`. This component references the response file that you created for use with a package.

Solaris Package Container Components

Containers are components that reference other components. Use the Package containers to create components that include a Package component and its corresponding Response File component. For more information about containers, see Chapter 1, "Plan and Component Development Concepts," in *Sun N1 Service Provisioning System 5.1 Plan and Component Developer's Guide*.

Package Container Component and Datastream Package Container Component

If you want to deploy a Solaris package with a response file, from the Solaris Common Tasks page, click the Package with Response File: Create Package with Response File Container link. This link enables you to create a container component of the type `com.sun.solaris#PackageWithResp`. When you create this container component, include the Package component and its corresponding Response File component in the container component.

If you want to deploy a Solaris datastream package with a response file, from the Solaris Common Tasks page, click the Datastream Package with Response File: Create Datastream Package with Response File Container link. This link allows you to create a container component of the type `com.sun.solaris#StreamPackageWithResp`. When you create the container component, include the Datastream Package component and its corresponding Response File component in the container component.

Package Container Components and Datastream Package Container components contain the same component variables and component procedures. The variables and procedures are described below.

Component Variables

`installPath` The path on the target host where the package file and the response file will be downloaded.

Component Procedures

`default: install` Use this procedure to install a package with its response file on a target system or systems.

`mark Only: install`	If you manually installed a package with its response file on a target system or systems but want the provisioning system database to be aware that the package is installed on those systems, use this procedure.
`default: uninstall`	Use this procedure to uninstall a package and its response file from a target system or systems.
`markOnly: uninstall`	If you manually uninstalled a package and its response file from a target system or systems but want the provisioning system database to be aware that the package has been uninstalled from those systems, use this procedure.

CHAPTER 6

Using the Solaris Plug-In to Deploy Zones

This chapter explains how to use the Solaris 10 zone specific components and plans that are provided with the Solaris Plug-In. The chapter contains the following information:

- "Introduction to Zones Partitioning" on page 37
- "Creating and Managing Solaris Zones" on page 38
- "Solaris Zone Component Types" on page 40
- "Solaris Zone Container Component" on page 41
- "Solaris Zone Plans" on page 42
- "Hosts and Solaris Zones" on page 42
- "Troubleshooting Solaris Zones" on page 43

Introduction to Zones Partitioning

The Solaris Zones partitioning technology is used to virtualize operating system services and to provide an isolated and secure environment for running applications. A *zone* is a virtualized operating system environment that is created within a single instance of the Solaris Operating System (OS). When you create a zone, you produce an application execution environment in which processes are isolated from the rest of the system. This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in other zones. Even a process that is running with superuser credentials cannot view or affect activity in other zones.

Zones can be used on any machine that is running the Solaris 10 OS. The total resource requirements of the application software running in all of the zones determines how many zones a single system can host effectively. For more information about Solaris Zones, see Part II, "Zones," in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*.

Creating and Managing Solaris Zones

The Common Tasks page of the software enables you to create and manage Solaris 10 zones.

▼ How to Define a Host as a Global Zone

A global zone is a host on which individual zones are defined. Before you can create any local zones, you must have at least one host defined to be in the host set `com.sun.solaris#global_zones`. This task explains how to define a host as a global zone.

Steps

1. **In the left control panel, under the Hosts Setup section, click Hosts.**
2. **Type the name of the host that you want to create as a global zone (container) in the Name field and click Create.**
3. **Choose `com.sun.solaris#global_zone` from the host type menu.**
4. **Modify values for any attributes as needed.**

   For example, you might want to change the connection type from RAW (the default) to SSH or SSL.

   Note – To use SSH for the local zone connection, you must set the port to 70000.

5. **Click the appropriate checkbox to include a Remote Agent or a Local Distributor on the physical host.**

▼ How to Create a Solaris Local Zone

Steps

1. **In the left control panel, under Common Tasks, click the Sun Solaris icon.**
2. **Click the Solaris Containers: Create and Manage link.**
3. **Click the Run action next to Create: Install.**
4. **Select the target host on which to create the local zone.**

Note – The target host must be a member of the `com.sun.solaris#global_zones` host set.

---

5. **Select the local zone host name from the Plan Variables settings.**

   If desired local zone names are not present, you can create new zone names. To create new zone names, follow these steps:

   a. **Click Select From List in the plan variable settings table.**

   b. **In the Variable Setting window, click Create Set**

   c. **Type the new zone set name in the New Set Name field.**

   d. **Click the box next to Zone Name prompt and type a new zone name.**

   Note – Zone names become network Zone Host Names during local zone creation. Zone Host Names should resolve to matching available local network IP addresses.

   ---

   e. **Determine if you want the local zone file system to be sparse or full.**

   A sparse local zone file system shares the `/lib`, `/sbin`, `/platform`, and `/usr` directories with the global zone. If the local zone will run applications that must install files into one of these directories, then install the local zone with a full file system. The full local zone file system does not share directories with the global zone.

   Note – The provisioning system installs the local zone file system as sparse by default. Install a full local zone file system by overriding the default.

   ---

6. **Click Run Plan (includes preflight).**

Using the CLI to Work With Solaris Zones

By default, the N1 Service Provisioning System command-line interface is located in the following directory:
`/opt/SUNWn1sps/N1_Grid_Service_Provisioning_system_5.0/cli/bin/cr_cli`

Solaris zone plug-in plans that run `Container` procedures are located in the folder `/com/sun/solaris/zones`.

EXAMPLE 6–1 Creating a Solaris Zone by Using the CLI

This example illustrates how you can use the N1 Service Provisioning System command-line interface to create a Solaris zone. The example contains three command segments. All commands use an example user name of `spsadmin` and password of `x1y2z3`.

The first command creates a host for global zone on `server10` and defines default values for the `local_zone_default_name`, `local_zone_connection_type`, and `local_zone_port` variables.

```
% cr_cli -cmd hdb.h.mod                          host modify
    -ID "NM:server10"                           target global host
    -tID "NM:com.sun.solaris#global_zone"       host type
    -attr "local_zone_default_name=server10_z1;
     local_zone_default_filesystem=FULL;
     local_zone_connection_type=RAW;
     local_zone_port=1131"                      local zone attributes
    -u spsadmin                                 user name
    -p x1y2z3                                   password
```

The second command creates a local zone using the default variables.

```
% cr_cli -cmd pe.p.run                          plan execution
    -PID "NM:/com/sun/solaris/Container-create"          create zone plan ID
    -tar "H:NM:server10"                        target global host
    -vs "+"                                     default var set "default set"
    -comp "+"                                   default component "latest build"
    -pto 6000                                   plan time 1 hr max [6000 secs]
    -nto 6000                                   call time 1 hr max [6000 secs]
    -u spsadmin                                 user name
    -p x1y2z3                                   user password
```

The third command activates the local zone.

```
% echo "server10_z1" | cr_cli -cmd pe.p.run     plan execution, std input is zone name
    -PID "NM:/com/sun/solaris/Container-activate"       attach zone plan ID
    -tar "H:NM:server10"                        target global host
    -f "-"                                      standard input args
    -pto 6000                                   plan time 1 hr max [6000 secs]
    -nto 6000                                   call time 1 hr max [6000 secs]
    -u spsadmin                                 user name
    -p x1y2z3                                   user password
```

Solaris Zone Component Types

Solaris containers have no specific component types. However, some specific host types are associated with Solaris containers. For information, see "Hosts and Solaris Zones" on page 42.

Solaris Zone Container Component

The Container component is the main component delivered by the Solaris plug-in for use with Solaris zones. The Container component is an untyped component.

The Container component includes several procedures. You can run these procedures directly from the N1 Service Provisioning System interface or from within execution plans.

Component Procedures

create:install	Creating a Container component creates a corresponding local Solaris 10 zone. The host target of this procedure is a Solaris 10 host that is running a provisioning system Remote Agent. The target host type must be set to `com.sun.solaris#global_zone`. The create procedure uses Solaris 10 `zonecfg` and `zoneadm` commands to configure and install a local Solaris 10 zone from the Solaris 10 global zone. After the installation, this procedure then installs a Remote Agent on the newly create local zone. A new host of type `com.sun.solaris#local_zone` appears on the host page.
attach:install	This procedure is similar to the create procedure, but it does not create the zone. Instead, the procedure attaches to an existing local zone on the host. This procedure creates a Container component in the provisioning system database and installs a remote agent on the existing local zone. The attach procedure is useful for times when a tool other than N1 Service Provisioning System created the zone. For example, you might use the attach procedure when a zone has been created directly through Solaris APIs or by another software management application.
markOnly:install	The markOnly:install procedure adds a Solaris Zone Container Component to the provisioning system database. The markOnly:install procedure does not create a local zone. The markOnly-install procedure does not install a Remote Agent.
delete:uninstall	The delete procedure uses the Solaris 10 `zonecfg` and `zoneadm` commands to remove the specified Solaris 10 local zone from the global zone host. The procedure also removes the corresponding Container component from the provisioning system database.

detach:uninstall	The detach procedure is similar to the delete procedure. This procedure removes the Container component from the provisioning system database and uninstalls the Remote Agent. However, the local zone is not removed from the global zone host. If the local zone is booted, it remains booted. If the local zone is halted, it remains halted.
markOnly:uninstall	The markOnly:uninstall procedure removes a Solaris Zone Container Component from the provisioning system database. The markOnly:uninstall procedure does not delete a local zone. The markOnly-delete procedure does not uninstall a Remote Agent.
activate	This procedure activates (boots) a local zone. You select the zone from the list of installed Container components on the targeted Solaris 10 host. The procedure uses the `zoneadm -z` *zonename* `boot` command. This procedure also initiates provisioning system host preparation for unprepared hosts.
deactivate	This procedure deactivates (halts) a local zone. You select the zone from the list of installed Container components on the targeted Solaris 10 host. The procedure uses the `zoneadm -z` *zonename* `halt` command.

Solaris Zone Plans

For each of the procedures described for the Container component, an associated plan is provided in the `/com/sun/solaris plan` folder. These plans are convenient to use with command-line shell calls, as shown in Example 6–1.

Hosts and Solaris Zones

The Solaris Plug-In includes a specific host set, a host search, and two host types for Solaris 10 zones.

- Global Zone host set – Any host on which you want to install a zone must be a member of the global zone host set. To ensure this relationship, set the host type of the target host to `com.sun.solaris#global_zone`.

- Global Zone host search – The global zones host search provides a search string that matches any host that is a global zone.
- Global Zone host type – Any physical host on which you want to install a zone must have a host type of `com.sun.solaris#global_zone`.
- Local Zone host type – When you install the `Container` component onto a global zone host, the provisioning system creates a host of type `com.sun.solaris#local_zone`.

Troubleshooting Solaris Zones

If you are experiencing problems managing your Solaris zones through the provisioning system, messages will likely appear on the screen to tell you that a problem exists.

For example, if you try to create a local zone for a target host that is not known to be a global zone, then you see a message similar to the following:

```
Problems encountered during plan run or preflight

The plan (or preflight) "/system/autogen/Container-inst-create-1098225529078"
finished with 1 failed host(s).

The specified target host "masterserver" was not in the required host set
"com.sun.solaris#global_zones" for component "/com/sun/solaris/Container".
```

Although the message in this example is easily deciphered, not all messages are this straightforward. To find out more information about the specific problem, follow the Details links.

Index

A

administration file, 31-33

C

component procedures
  datastream package, 33
  datastream package container component, 34-35
  package, 33
  package container component, 34-35
  patch, 25
  zones, 41-42
component variables
  datastream package, 31-33
  datastream package container component, 34
  package, 31-33
  package container component, 34
  patch, 25
config.properties file, 22
configuring the plug-in, 22
creating
  datastream package component, 28-29
  datastream package container component, 29-30
  local zone, 38-39
  package component, 28-29
  package container component, 29-30
  patch component, 24
  response file component, 29-30
  zone using CLI, 39-40

D

datastream package
  component procedures, 33
  component type, 31-33
  component variables, 31-33
  container component
    component procedures, 34-35
    component variables, 34
    creating, 29-30
    description, 34-35
  creating component, 28-29
  definition of, 27-28
  installing
    with response file, 29-30
    without a response file, 28-29
deployment server requirements, 13

F

file system package
  component type, 31-33
  definition of, 27-28

G

global zone host search, 42-43
global zone host set, 42-43
global zone host type, 42-43
global zone in SPS, 38

H

host type
  global zone, 42-43
  local zone, 42-43

I

identifying a global zone, 38
importing the plug-in, 18-19

J

JAR file, 17
Java Archive, *See* JAR file

L

local zone host type, 42-43
local zone in SPS
  creating, 38-39
  upgrading, 19-21

P

package
  component
    creating, 28-29
  component procedures, 33
  component type, 31-33
  component variables, 31-33
  container component
    component procedures, 34-35
    component variables, 34
    creating, 29-30
    description, 34-35
  datastream, 27-28
  definition of, 27-28
  deploy, 28-30
  file package, 27-28
  installing
    with response file, 29-30
    without a response file, 28-29
  parts, 12
  resource file, 27-28
patch
  component procedures, 25
  component type, 24-25
  component variables, 25
  create component, 24
  definition of, 23
  parts, 12
plug-in files
  importing, 18-19
  location of, 17
  upgrading, 18-19

R

requirements
  for Master Server, 13
  for target hosts, 13-14
resource file, definition of, 27-28
response file, component type, 33

S

Solaris plug-in
  configuration, 22
  importing, 18-19
  Master Server requirements, 13
  parts, 11-13
  target host requirements, 13-14
  upgrading, 18-19

T

target host requirements, 13-14
timeout value, 22

U

upgrading
  local zones, 19-21
  the plug-in, 18-19
upgrading the plug-in, upgrading local zones, 19-21
using CLI, create a zone, 39-40

Z

zones
and CLI, 39-40
container component, 41-42
container component procedures, 41-42
definition of, 37
global, 38
local, 38-39
parts, 12-13