



# Sun N1 Service Provisioning System User's Guide for Solaris Plug-In 4.0



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

Part No: 819-4902-10  
April 2006

Copyright 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

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.

This distribution may include materials developed by third parties.

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, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, N1, Java, 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.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, N1, Java, 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.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

# Contents

---

<b>Preface</b> .....	7
<b>1 Overview of Solaris Plug-In</b> .....	11
Purpose of the Solaris Plug-In .....	11
What the Solaris Plug-In Includes .....	11
Solaris Patches .....	11
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
<b>2 Release Notes</b> .....	15
Installation Issues .....	15
Runtime Issues .....	15
Component Installation Paths Must Be Unique (6349681) .....	15
Solaris Plug-In Cannot Remove Container Component If Zone Is Removed Manually .....	15
<b>3 Installing and Configuring the Solaris Plug-In</b> .....	17
Acquiring the Solaris Plug-In .....	17
Adding the Solaris Plug-In for Solaris .....	18
▼ To Add the Solaris Plug-In for Solaris .....	18
Adding the Solaris Plug-In for Linux .....	18
▼ To Add the Solaris Plug-In for Linux .....	18
Adding the Solaris Plug-In for Windows .....	18
▼ To Add the Solaris Plug-In for Windows .....	19
Importing the Solaris Plug-In to the Sun N1 Service Provisioning System .....	19
▼ How to Import the Solaris Plug-In Using the Browser Interface .....	19

▼ How to Import the Solaris Plug-In Using the CLI .....	20
Upgrading Considerations .....	20
Upgrading Existing Local Zones .....	20
▼ How to Upgrade Global Zones .....	20
▼ How to Upgrade a Local Zone that is Running .....	20
▼ How to Upgrade a Local Zone that is Not Running .....	21
Accessing Previous Versions of a Component .....	22
▼ How to Access Previous Versions of a Component .....	22
Customizing the Solution for Your Environment .....	23
▼ How to Customize the <code>config.properties</code> File .....	23
Patching the Solaris Plug-In .....	23
<b>4 Using the Solaris Plug-In to Deploy Patches .....</b>	<b>25</b>
Introduction to Solaris Patches .....	25
Deploying Solaris Patches .....	25
Solaris Patch Component Type .....	26
▼ How to Deploy a Solaris Patch from the Solaris Patch Component Type .....	26
Component Variables .....	27
Component Procedures .....	27
Solaris Patch Component .....	27
▼ How To Install Solaris Patches From the Solaris Patch Component .....	28
▼ How To Uninstall Solaris Patches From the Solaris Patch Component .....	29
<b>5 Using the Solaris Plug-In to Deploy Packages .....</b>	<b>31</b>
Introduction to Solaris Packages .....	31
Deploying Solaris Packages .....	32
▼ How to Deploy a Solaris Package or Datastream Package Without a Response File .....	32
▼ How to Deploy a Solaris Package or DataStream Packages With a Response File .....	33
Solaris Package Component Types .....	34
Package and Datastream Package Component Types .....	34
Response File Component .....	36
Solaris Package Container Component Types .....	37
Package Container Component and Datastream Package Container Component .....	37
Solaris Package Component .....	38
▼ How To Install Solaris Packages from Solaris Package Component .....	38
▼ How to Uninstall a Solaris Package Component .....	39

Solaris Datastream Package Component .....	39
▼ How to Install Solaris Datastream Packages From Solaris Datastream Package Component ....	40
▼ How to Uninstall a Solaris Datastream Package Component .....	41
<b>6 Using the Solaris Plug-In to Deploy Zones .....</b>	<b>43</b>
Introduction to Zones Partitioning .....	43
Creating and Managing Solaris Zones .....	44
▼ How to Define a Host as a Global Zone .....	44
▼ How to Create a Solaris Local Zone .....	44
Using the CLI to Work With Solaris Zones .....	46
Solaris Zone Component Types .....	47
Solaris Zone Container Component .....	47
Component Procedures .....	47
Solaris Zone Plans .....	48
Hosts and Solaris Zones .....	49
Troubleshooting Solaris Zones .....	49
<b>Index .....</b>	<b>51</b>



# Preface

---

The *Sun N1 Service Provisioning System User's Guide and Release Notes for the Solaris Plug-In 4.0* 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 4.0* 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 Sun N1 Service Provisioning System software, read the following books:

- *Sun N1 Service Provisioning System 5.2 System Administration Guide*
- *Sun N1 Service Provisioning System 5.2 Operation and Provisioning Guide*
- *Sun N1 Service Provisioning System 5.2 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

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

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

## Typographic Conventions

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

TABLE P-1 Typographic Conventions

Typeface	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% you have mail.</code>
<b>AaBbCc123</b>	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> 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> . A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file. <b>Note:</b> Some emphasized items appear bold online.



# Shell Prompts in Command Examples

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

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	machine_name%
C shell for superuser	machine_name#
Bourne shell and Korn shell	\$
Bourne shell and Korn shell for superuser	#



# ◆ ◆ ◆ CHAPTER 1

## Overview of Solaris Plug-In

---

This chapter explains general information about using Sun 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 Sun 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 `/opt/SUNWn1sps/plugins/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

- |   |  |
|---|--|
| <code>Container</code> component                      | Main component used to create and manage zones. For more information, see <a href="#">“Solaris Zone Container Component”</a> on page 47. |
| <code>container_util</code> component                 | Component that contains utility scripts for zones  |
| <code>Container-create</code> plan                    | Plan that creates zones  |
| <code>Container-attach</code> plan                    | Plan that attaches zones   |
| <code>Container-delete</code> plan                    | Plan that deletes zones  |
| <code>Container-detach</code> plan                    | Plan that detaches zones   |
| <code>Container-activate</code> plan                  | Plan that activates zones  |
| <code>Container-deactivate</code> plan                | Plan that deactivates zones  |
| <code>com.sun.solaris#global_zones</code> host set    | Host set for global zones  |
| <code>com.sun.solaris#global_zones</code> host search | Host search for global zones   |
| <code>com.sun.solaris#global_zone</code> host type    | Host type for global zones   |
| <code>com.sun.solaris#local_zone</code> host type     | Host type for local zones  |

# Requirements for Using the Solaris Plug-In

## Requirements for Master Server



---

**Caution** – The Solaris Plug-In 4.0 runs in the Sun N1 Service Provisioning System 5.2 environment. Do not install the Solaris Plug-In 4.0 on any other version of the Sun N1 Service Provisioning System.

---

To deploy Solaris patches, packages, and zones, make sure that the following two parts of the Sun 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<sup>®</sup> 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.



## 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

#### **Component Installation Paths Must Be Unique (6349681)**

If you do not change the `installPath` component variable for each installation, only the most recently installed component is available to be uninstalled. This issue applies to the following components:

- Components created by using the Patch component type
- Components created by using the Package component type
- Components created by using the Datastream Package component type

**Workaround:** Provide a unique `installPath` component variable for each component.

#### **Solaris Plug-In Cannot Remove Container Component If Zone Is Removed Manually**

If a zone that was created by installing a Solaris component from the Solaris plug-in is manually deleted, the SPS component cannot be uninstalled in SPS.

The following error messages are displayed:

Problems encountered during plan run or preflight

The plan (or preflight) `"/system/autogen/Container-unst-markOnly-1141844440690"` finished with 1 failed host(s). (017034)

The targetable component host cannot be uninstalled as it contains one or more installed components. (603013)

**Workaround:** Manually recreate the Solaris Zone on the Remote Agent.



# Installing and Configuring the Solaris Plug-In

---

This chapter contains the following information:

- “Acquiring the Solaris Plug-In” on page 17
- “Importing the Solaris Plug-In to the Sun N1 Service Provisioning System” on page 19
- “Upgrading Considerations” on page 20
- “Customizing the Solution for Your Environment” on page 23

## Acquiring the Solaris Plug-In

Acquiring the Solaris containers plug-in is a two-step process. First, you must add the package file that contains the Solaris containers plug-in JAR file to your system. Then you must import the Solaris containers plug-in JAR file.

The Solaris Plug-In is packaged as a *plug-in* to the Sun 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 5.2 Supplement CD or from the Sun Download Center.

Once the package file is added to your system, the Solaris Plug-In is available for import from two different JAR files. Choose the correct file depending on your situation.

- If you are importing the Solaris Plug-In for the first time, acquire the `com.sun.solaris_4.0.jar` file.
  - If you have already imported the previous version of the Solaris Plug-In, acquire the `com.sun.solaris_3.0_4.0.jar` file.
1. Add the file containing the JAR file.
    - “Adding the Solaris Plug-In for Solaris” on page 18
    - “Adding the Solaris Plug-In for Linux” on page 18
    - “Adding the Solaris Plug-In for Windows” on page 18
  2. Import the JAR file – “Importing the Solaris Plug-In to the Sun N1 Service Provisioning System” on page 19

## Adding the Solaris Plug-In for Solaris

The Solaris plug-in is contained in the SUNWspssol package.

### ▼ To Add the Solaris Plug-In for Solaris

- 1 In a terminal window, become superuser.
- 2 Move to the directory containing the plug-in package.
- 3 Type the following command and press Return.

```
# pkgadd -d . SUNWspssol
```

The standalone JAR file is in the `/opt/SUNWn1sps/plugins/com.sun.solaris/` directory. The upgrade JAR file is in the `/opt/SUNWn1sps/plugins/com.sun.solaris/Upgrade` directory.

## Adding the Solaris Plug-In for Linux

The Solaris plug-in is contained in the SUNWspssol.rpm file.

### ▼ To Add the Solaris Plug-In for Linux

- 1 In a terminal window, become superuser.
- 2 Move to the directory containing the SUNWspssol.rpm file.
- 3 Type the following command and press Return.

```
# rpm -i package_directory/SUNWspssol
```

The standalone JAR file is in the `/opt/sun/N1_Service_Provisioning_System/plugins/com.sun.solaris/` directory. The upgrade JAR file is in the `/opt/sun/N1_Service_Provisioning_System/plugins/com.sun.solaris/Upgrade` directory.

## Adding the Solaris Plug-In for Windows

The Solaris plug-in is contained in the SUNWspssol.msi file.

## ▼ To Add the Solaris Plug-In for Windows

- 1 Move to the directory containing the `SUNWspssol.msi` file.
- 2 Double-click the `SUNWspssol.msi` file.

The Installer GUI starts. The JAR file is copied to the `C:\Program Files\N1 Service Provisioning System\plugins\com.sun.solaris` directory.

# Importing the Solaris Plug-In to the Sun N1 Service Provisioning System

To make a given plug-in known to the Sun 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 5.2 System Administration Guide*.



---

**Caution** – The Solaris Plug-In 4.0 runs in the Sun N1 Service Provisioning System 5.2 environment. Do not install the Solaris Plug-In 4.0 on any other version of the Sun N1 Service Provisioning System.

---

- 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 Navigate to the location of the JAR file.
  - If you are importing the Solaris Plug-In for the first time, select the `com.sun.solaris_4.0.jar` file.
  - If you have already imported a previous version of the Solaris Plug-In, select the `com.sun.solaris_3.0_4.0.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.

### ► 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_4.0.jar`.
- If you have already imported the previous version of the Solaris Plug-In, *plugin-filename* is `com.sun.solaris_3.0_4.0.jar`.

# Upgrading Considerations

## Upgrading Existing Local Zones

If you have created global and local zones by using the Solaris Plug-In in the Sun Sun N1 Service Provisioning System 5.0, after you migrate the Master Server to the Sun N1 Service Provisioning System 5.2 and you import the Solaris Plug-In 4.0, you must upgrade the zones. You can then control the upgraded zones with the Solaris Plug-In 4.0.

## ▼ How to Upgrade Global Zones

- 1 In the Common Tasks section of the browser interface main window, click the Solaris icon.
- 2 Click the Solaris Container: Create and Manage link.
- 3 In the Referenced 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 5.2, upgrade the Remote Agents. For instructions see “Upgrading Remote Agents and Local Distributors” in *Sun N1 Service Provisioning System 5.2 Installation Guide*.

- 1 In the **Common Tasks** section of the browser interface main window, click the **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 **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.

- 1 In the **Common Tasks** section of the browser interface main window, click the **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 3.0 version to the 4.0 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:

**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

- 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/SUNWnlsp/N1_Service_Provisioning_System_5.2/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 Sun 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.**

## Patching the Solaris Plug-In

Check the SunSolve (<http://sunsolve.sun.com>) site for available patches for the Solaris plug-in. To apply the patch, follow the instructions in the patch README file.





# Using the Solaris Plug-In to Deploy Patches

---

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

- “Introduction to Solaris Patches” on page 25
- “Deploying Solaris Patches” on page 25
- “Solaris Patch Component Type” on page 26
- “Solaris Patch Component” on page 27

## 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 Sun 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 and Updates (Overview),” in *System Administration Guide: Basic Administration*.

## Deploying Solaris Patches

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

There are two methods by which you can install a patch:

- “Solaris Patch Component Type” on page 26
- “Solaris Patch Component” on page 27

# Solaris Patch Component Type

To deploy Solaris patches using the Sun 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 5.2 Plan and Component Developer’s Guide*.

- “How to Deploy a Solaris Patch from the Solaris Patch Component Type” on page 26
- “Component Variables” on page 27
- “Component Procedures” on page 27

## ▼ How to Deploy a Solaris Patch from the Solaris Patch Component Type

**Before You Begin** Obtain the patch that you want to deploy.



---

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

---

- 1 In the left control panel, under Common Tasks, click the Solaris link.**  
The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
- 2 Click the Solaris Patch Component Type: 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 27.
- 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 27.
- 8 Complete the plan run page to run the procedure.**

## Component Variables

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

---

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

---

## Component Procedures

<code>default: install</code>	Use this procedure to install a patch on a target system or systems.
<code>markOnly: install</code>	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.
<code>default: uninstall</code>	Use this procedure to uninstall a patch from a target system or systems.
<code>markOnly: uninstall</code>	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.

## Solaris Patch Component




---

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

---

- [“How To Install Solaris Patches From the Solaris Patch Component” on page 28](#)
- [“How To Uninstall Solaris Patches From the Solaris Patch Component” on page 29](#)

## ▼ How To Install Solaris Patches From the Solaris Patch Component

- 1 In the left control panel, under Common Tasks, click the Solaris link.**

The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
- 2 Click the Solaris Patch Component: Install link.**
- 3 Click Select From List in the Plan Parameters table.**
- 4 In the Variable Setting window, click Create Set.**
- 5 Type the new Set Name.**
- 6 Select the `installPath` check box and type a unique value.**

The `installPath` value must be unique. If you do not change the `installPath` component variable for each installation, only the most recently installed component is available for uninstallation.
- 7 Select the `rootDirectory` check box and type the root directory for deployment.**
- 8 Select the `validateFiles` check box and type `true` to set validation of installed files.**

This refers to the `-u` option of the `patchadd` command.
- 9 Select the `patchIDs` check box and type enter space-separated patch IDs.**
- 10 Select the `patchLocation` and type the absolute path to where the patch IDs are located.**
- 11 Click Save and Select.**
- 12 Click Select From List for the Target Host.**
- 13 Click the Add Hosts to Main Window button.**
- 14 (Optional) Select the `markOnly` installation check box.**
- 15 Click Run Plan (Includes Preflight).**

## ▼ How To Uninstall Solaris Patches From the Solaris Patch Component

- 1 In the left control panel, under **Common Tasks**, click the **Solaris** link.

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

- 2 Click the **Solaris Patch Component: Uninstall** link.

- 3 Select a host install path.

- 4 Click **Run Selected Installation**.

- 5 Select plan variables.

- Force the patch remove (-f option of patchrm)
- markonly uninstall

- 6 Click **Run Plan (Includes Preflight)**.



# 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 31
- “Deploying Solaris Packages” on page 32
- “Solaris Package Component Types” on page 34
- “Solaris Package Container Component Types” on page 37
- “Solaris Package Component” on page 38
- “Solaris Datastream Package Component” on page 39

## 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 Sun 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 Sun 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 Sun 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 32
- “How to Deploy a Solaris Package or DataStream Packages With a Response File” on page 33

## ▼ 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 Sun N1 Service Provisioning System to install the package.

**Before You Begin** Obtain the package that you want to install.



---

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

---

- 1 In the left control panel, under Common Tasks, click the Solaris link.**  
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 Component Type: Create Package Component link.
  - If you want to deploy a Solaris datastream package, click the Solaris Datastream Package Component Type: 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 35.

- 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 36.

## ▼ 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 Sun N1 Service Provisioning System does not check for package dependencies. Be sure to install packages in the proper sequence to account for package dependencies.

---

- 1** In the left control panel, under **Common Tasks**, click the **Solaris** link.

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 Component Type: Create Package Component link.
- If you want to deploy a Solaris datastream package, click the Solaris Datastream Package Component Type: Create Datastream Package Component link.

- 3** In the left control panel, under **Common Tasks**, click the **Solaris** link.

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 Solaris link.**

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 37](#).

**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 37](#).

## Solaris Package Component Types

### Package and Datastream Package Component Types

Use the Package component types 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 5.2 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

<code>installName</code>	The name to use for the package when it is installed. The default value is the name of the package component.
<code>installPath</code>	The path on the target host where the package will be downloaded.
<code>installRoot</code>	The path on the target host where the package will be installed.
<code>pkgName</code>	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
```

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

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

<code>default: install</code>	Use this procedure to install a package on a target system or systems.
<code>mark Only: install</code>	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.
<code>default: uninstall</code>	Use this procedure to uninstall a package from a target system or systems.
<code>markOnly: uninstall</code>	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 Sun 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 Component Types

Containers are components that reference other components. Use the Package container type 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 5.2 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.

<code>mark Only: install</code>	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.
<code>default: uninstall</code>	Use this procedure to uninstall a package and its response file from a target system or systems.
<code>markOnly: uninstall</code>	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.

## Solaris Package Component



---

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

---

- [“How To Install Solaris Packages from Solaris Package Component” on page 38](#)
- [“How to Uninstall a Solaris Package Component” on page 39](#)

### ▼ How To Install Solaris Packages from Solaris Package Component

- 1 In the Common Tasks section of the browser interface main window, click the Solaris icon.**  
The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
- 2 Click the Solaris Package Component: Install link.**
- 3 Click Select From List in the Plan Parameters table.**
- 4 In the Variable Setting window, click Create Set.**
- 5 Type the new Set Name.**
- 6 Select the `installPath` check box and type a unique value.**  
The `InstallPath` value must be unique. If you do not change the `installPath` component variable for each installation, only the most recently installed component is available for uninstallation.
- 7 (Optional) Select the `installRoot` check box and type a relative root path.**  
This is used as the `-R` option to the `pkgadd` command.

- 8 Select the `pkgNames` check box and type space-separated package names.
- 9 Select the `pkgLocation` check box and type the absolute path for the packages location.
- 10 (Optional) Override the installation administration file, admin file, parameters by select the check boxes next to admin file component variables and typing the values.
- 11 Click Save and Select.
- 12 Click the Select From List link for the target host.
- 13 Select a host and click the Add Hosts to Main Window button.
- 14 (Optional) Select the `markOnly Installation` check box.
- 15 Click Run Plan (Includes Preflight).

## ▼ How to Uninstall a Solaris Package Component

- 1 In the Common Tasks section of the browser interface main window, click the Solaris icon.  
The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
- 2 Click the Solaris Package Component: Uninstall link.
- 3 Select a host install path.
- 4 Click Run Selected Installations.
- 5 Select plan variable.
  - `markonly uninstall`
- 6 Click Run Plan (Includes Preflight).

## Solaris Datastream Package Component



---

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

---

- “How to Install Solaris Datastream Packages From Solaris Datastream Package Component” on page 40
- “How to Uninstall a Solaris Datastream Package Component” on page 41

## ▼ How to Install Solaris Datastream Packages From Solaris Datastream Package Component

- 1 In the Common Tasks section of the browser interface main window, click the Solaris icon.
- 2 Click the Solaris Datastream Package Component: Install link
- 3 Click the Select From List link in the Plan Parameters table.
- 4 In the Component Variables window, click Create Set.
- 5 Type the new Set Name.
- 6 Select the `installPath` check box and type a unique value.  
The `installPath` value must be unique. If you do not change the `installPath` component variable for each installation, only the most recently installed component is available for uninstallation.
- 7 (Optional) Select the `installRoot` check box and type a relative root path.  
This is used as the `-R` option to the `pkgadd` command.
- 8 Select the `pkgNames` check box and type space-separated package names.
- 9 Select the `pkgLocation` check box and type the absolute path of the datastream package location.
- 10 (Optional) Override the installation administration file, admin file, parameters by clicking boxes next to admin file component variables and entering the values
- 11 Select the `streamPkgName` check box and type the name of the datastream package.  

---

Note – The plug-in supports only one datastream package name.

---
- 12 Click Save and Select.
- 13 Click the Select From List link for the Target Host.
- 14 Select a host and click the Add Hosts to Main Window button.



- 15 (Optional) Select the `markOnly` installation check box.
- 16 Click Run Plan (Includes Preflight).

## ▼ How to Uninstall a Solaris Datastream Package Component

- 1 In the Common Tasks section of the browser interface main window, click the Solaris icon.  
The Common Tasks for the Solaris Plug-In appear in the right panel of the browser interface.
- 2 Click the Solaris Datastream Package Component: Uninstall link
- 3 Select host install path.
- 4 Click Run Selected Installations.
- 5 Select plan variable.
  - `markonly uninstall`
- 6 Click Run Plan (Includes Preflight).



# 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 43
- “Creating and Managing Solaris Zones” on page 44
- “Solaris Zone Component Types” on page 47
- “Solaris Zone Container Component” on page 47
- “Solaris Zone Plans” on page 48
- “Hosts and Solaris Zones” on page 49
- “Troubleshooting Solaris Zones” on page 49

## 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.

- 1 In the left control panel, under the Hosts Setup section, click Hosts.
- 2 Designate the host system by choosing one of the following options:
  - User provided Solaris 10 host.
    - a. Type a Solaris 10 host name in the Host field.
    - b. Click Create.
  - Solaris 10 host from the host table.
    - a. Select a Solaris 10 host from the host table.
    - b. Click Edit.
- 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.
- 6 Click Save.

## ▼ How to Create a Solaris Local Zone

**Before You Begin** If there is no Name Server, you must update the `/etc/hosts` file on the Master Server with the `hostname:ip_address` of the local zone on the Remote Agent.

- 1 In the left control panel, under Common Tasks, click the Solaris link.
- 2 Click the Solaris Container: Create and Manage link.

- 3 Click the Run action next to Install: Create.
- 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 parameters 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 variable set name.

---

**Note** – Zone names become network Zone Host Names during local zone creation. Local Zone Host Names should resolve to local network IP address.

---

- 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.

---

- f. (Optional) If you need to add the logical network interfaces into the local zone, type values into the `zoneInterfaceDetails` component variable.

Each entry consists of three fields in the following format: *network interface,IPAddress/netmask*. Each entry is separated by a semicolon. For example:  
`hme0,123.123.123.123/24;eri0,124.124.124.124/8`

- g. (Optional) If you need to mount global zone file systems in the local zone, type values into the `zoneFsLayout` component variable.

Each file system must be separated by a semicolon. For example, `dir=/usr/local special=/opt/local raw=/dev/rdisk/c0t0d0s7 type=lofs [ro,nodevices];dir=/opt/mystuff special=/empty type=lofs ro`

For more information, see the `zonecfg(1M)` man page.

- 6 Click Save and Select.
- 7 Click Select From List for the target host.
- 8 Select one of Current Found Hosts.
- 9 Click Add Hosts To Main Window.
- 10 Click Run Plan (includes preflight).

## Using the CLI to Work With Solaris Zones

By default, the Sun N1 Service Provisioning System command-line interface is located in the following directory: `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/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 Sun 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]
```

**EXAMPLE 6-1** Creating a Solaris Zone by Using the CLI (Continued)

```

-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 49.

## 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 Sun 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 <code>com.sun.solaris#global_zone</code> . The create procedure uses Solaris 10 <code>zonecfg</code> and <code>zoneadm</code> 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 <code>com.sun.solaris#local_zone</code> 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 Sun 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 <code>zonecfg</code> and <code>zoneadm</code> 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 <code>zoneadm -z zonename boot</code> 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 <code>zoneadm -z zonename halt</code> 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, 35-36

## C

component procedures

- datastream package, 36
- datastream package container component, 37-38
- package, 36
- package container component, 37-38
- patch, 27
- zones, 47-48

component variables

- datastream package, 35-36
- datastream package container component, 37
- package, 35-36
- package container component, 37
- patch, 27

config.properties file, 23

configuring the plug-in, 23

creating

- datastream package component, 32-33
- datastream package container component, 33-34
- local zone, 44-46
- package component, 32-33
- package container component, 33-34
- patch component, 26
- response file component, 33-34
- zone using CLI, 46-47

## D

datastream package

- component procedures, 36
  - component type, 34-36
  - component variables, 35-36
  - container component
    - component procedures, 37-38
    - component variables, 37
    - creating, 33-34
    - description, 37-38
  - creating component, 32-33
  - definition of, 31
  - installing
    - with response file, 33-34
    - without a response file, 32-33
- deployment server requirements, 13

## F

file system package

- component type, 34-36
- definition of, 31

## G

- global zone host search, 49
- global zone host set, 49
- global zone host type, 49
- global zone in SPS, 44

**H**

- host type
  - global zone, 49
  - local zone, 49

**I**

- identifying a global zone, 44
- importing the plug-in, 19-20

**J**

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

**L**

- local zone host type, 49
- local zone in SPS
  - creating, 44-46
  - upgrading, 20-22

**P**

- package
  - component
    - creating, 32-33
    - component procedures, 36
    - component type, 34-36
    - component variables, 35-36
  - container component
    - component procedures, 37-38
    - component variables, 37
    - creating, 33-34
    - description, 37-38
  - datastream, 31
  - definition of, 31
  - deploy, 32-34
  - file package, 31
  - installing
    - with response file, 33-34
    - without a response file, 32-33

**package (Continued)**

- parts, 12
  - resource file, 31
- patch
    - component procedures, 27
    - component type, 26-27
    - component variables, 27
    - create component, 26
    - definition of, 25
    - parts, 11
  - plug-in files
    - importing, 19-20
    - location of, 17
    - upgrading, 19-20

**R**

- requirements
  - for Master Server, 13
  - for target hosts, 13
- resource file, definition of, 31
- response file, component type, 36-37

**S**

- Solaris plug-in
  - configuration, 23
  - importing, 19-20
  - Master Server requirements, 13
  - parts, 11-12
  - target host requirements, 13
  - upgrading, 19-20

**T**

- target host requirements, 13
- timeout value, 23

**U**

- upgrading
  - local zones, 20-22

---

upgrading (Continued)  
  the plug-in, 19-20  
upgrading the plug-in, upgrading local zones, 20-22  
using CLI, create a zone, 46-47

## **Z**

zones  
  and CLI, 46-47  
  container component, 47-48  
  container component procedures, 47-48  
  definition of, 43  
  global, 44  
  local, 44-46  
  parts, 12

