



Sun Open Telecommunications Platform 2.0 Developer's Guide



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

Part No: 820-3065
February 2008

Copyright 2008 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, 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 2008 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.

Certains 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, 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

- Preface7**
- 1 Supported Solution Topologies 13**
 - Solution Topologies 13
 - One Provisioning Server for Each Sun OTP Instance 13
 - One Provisioning Server for Each Tier 14
 - One Provisioning Server for Each Solution Instance 15
- 2 Developing Applications Using Sun OTP 17**
 - Interfaces Exposed by Sun OTP 17
 - Sun OTP Protocols 18
- 3 Sun OTP Tools for Application Development 21**
 - Tools for Application Development 21
- 4 Sun OTP Components23**
 - N1 SPS Components 23
 - N1 SPS Plans 25
 - Order of Installing N1 SPS Plans 26
 - Plan Re-runs 26
 - Plan Configuration Files 27
 - Utility Plans 30
 - Sun OTP AHE Plug-in 30
 - Customizing Sun OTP Deployment 30
 - Examples of Solution Specific Steps 31

5	Adapting Applications for Sun OTP	33
	Developing N1 SPS Plans for NEP Application Provisioning	33
	Calling Existing Application Installers from N1 SPS	33
	Moving an Existing Application Installer Code to an SPS Component	34
	Using Native N1 SPS Provisioning Features	36
	Developing Sun Cluster HA Agents for NEP Application Management	38
	Developing Agents Using Agent Builder	38
	Developing Agents Using Scripts and C/C++	38
6	Installing Sun OTP As Part of Existing Solution Deployment	39
	Calling N1 SPS Plans From Shell Scripts	39
	Examples	40
	Calling N1 SPS Plans From C/C++	41
	Calling N1 SPS Plans From Java	43
	Using N1 SPS From JumpStart	45
	Using N1 SPS From JET	46
7	Using Sun OTP for End-to-End Solution Deployment	47
	Stages in Using Sun OTP for End-to-End Solution Deployment	47
	Deployment Methods	49
8	Using Sun OTP Security Services	51
	Securing Web Applications With Sun OTP	51
	Configuring Policy Agent	51
	▼ To Configure Policy Agent for Web Server 6.1 on Solaris 10 11/06	52
	Integrating Web Applications Into Sun OTP Web SSO Environment	54
	Web Applications Without Any Authentication Implementation	54
	Web Applications With Their Own Authentication Implementation	54
	Hardening Sun OTP and Hosted NEP Applications	56
9	Examples of End-to-End Solution Deployment	57
	Prerequisites for Sun OTP Installation	57
	Prerequisite Tasks for Sun OTP Installation	58
	▼ To Install Solaris OS from the Bootable DVD	58

▼ To Prepare the Sun OTP Hosts	59
▼ To Set up the Self-Contained Sun OTP Provisioning Server	60
Installing Sun OTP Using GUI Installation	61
▼ To Install Sun OTP on a Clustered System	61
▼ To Configure the Quorum Disk on a Two-Host Cluster	76
Installing Sun OTP Using Command-Line Installation	77
▼ To Install Sun OTP on a Clustered System	77
 A Sun OTP Plan Worksheet	 81
Sun OTP Plan Settings Description	81
Sun OTP Plan Worksheet	85
 B Sun OTP Log Files	 87
Log Files	87
OTP Installation Log Files	87
Log File Locations	87
 Index	 89

Preface

Sun Open Telecommunications Platform 2.0 Developer's Guide describes how to develop and deploy applications using Sun™ OTP.

The following topics are discussed:

- [“What is Sun OTP?” on page 7](#)
- [“Target Audience” on page 9](#)
- [“Component Product Mapping” on page 10](#)
- [“Sun OTP Documentation Set” on page 11](#)
- [“Sun Welcomes Your Comments” on page 12](#)

What is Sun OTP?

Sun Open Telecommunications Platform (Sun OTP) provides integrated high availability services, system management services, application provisioning services, and security services that enable you to develop, deploy, host, and secure Network Equipment Provider (NEP) applications.

Sun OTP version 2.0 provides the following services:

- [“Provisioning Service” on page 7](#)
- [“Management Service” on page 8](#)
- [“Availability Service” on page 8](#)
- [“Security Service” on page 9](#)

Provisioning Service

Provisioning service consists of platform and application provisioning services.

Platform Provisioning

The platform provisioning service enables end-to-end provisioning of Sun OTP compute elements, including bare metal and firmware provisioning, operating system provisioning, and provisioning of Sun OTP software components.

Application Provisioning

The application provisioning service enables end-to-end provisioning of (NEP) applications, including initial application deployment, application upgrade, and application patching. Application provisioning services are capable of deploying applications on a single system, or on a group of systems that follow a set of defined grouping semantics. These services are also capable of deploying both single and multi-tier applications.

The following operations are supported by the application provisioning service:

- Creating application deployment descriptions
- Modifying application deployment descriptions
- Deleting application deployment descriptions
- Provisioning applications on hosts
- Provisioning a multi-tier application
- Inspecting deployed software on hosts at a specific point
- Removing applications from hosts
- Removing a multi-tier application
- Rolling back to previous version of applications
- Querying deployed applications on hosts

Management Service

Management service consists of platform management and application management services.

Platform Management

The platform management service enables monitoring and managing the Sun OTP compute elements. This includes monitoring and managing the bare metal hardware and deployed operating system instances. The platform management service can manage both stand-alone systems and two or more systems grouped together into an administrative group.

Application Management

The application management service enables management of NEP applications. Supported operations include application health monitoring, failure recovery and migration from one Sun OTP instance to another.

Availability Service

The availability service consists of platform availability and application availability services.

Platform Availability

The platform availability service enables availability of the Sun OTP compute elements.

Application Availability

The application availability service enables basic lifecycle and availability management of NEP applications. Supported lifecycle operations include registration, activation, and deactivation of applications.

The following operations are supported by the application lifecycle and availability management services:

- Creating application manifests
- Modifying application manifests
- Deleting application manifests
- Creating application dependencies
- Modifying application dependencies
- Deleting application dependencies
- Registering applications
- Starting applications
- Stopping applications
- Querying application state
- Migrating applications from host A to host B
- Failing over applications from host A to host B
- Restarting applications on host X

Security Service

The security service is used to secure NEP applications by authenticating Web applications through a Web Single Sign-On (Web SSO) feature. Once you log into a web-based administration interface in Sun OTP, you can access the other web-based administration interfaces without any reauthentication. Additionally, you can use the Solaris Security Toolkit (SST) driver to harden the Sun OTP application hosting environment (AHE) to improve the overall network security.

Target Audience

OEM developers who wish to install Sun OTP in their development environment and develop, deploy applications using Sun OTP.

Component Product Mapping

The following figure shows the components that are part of Sun OTP 2.0.

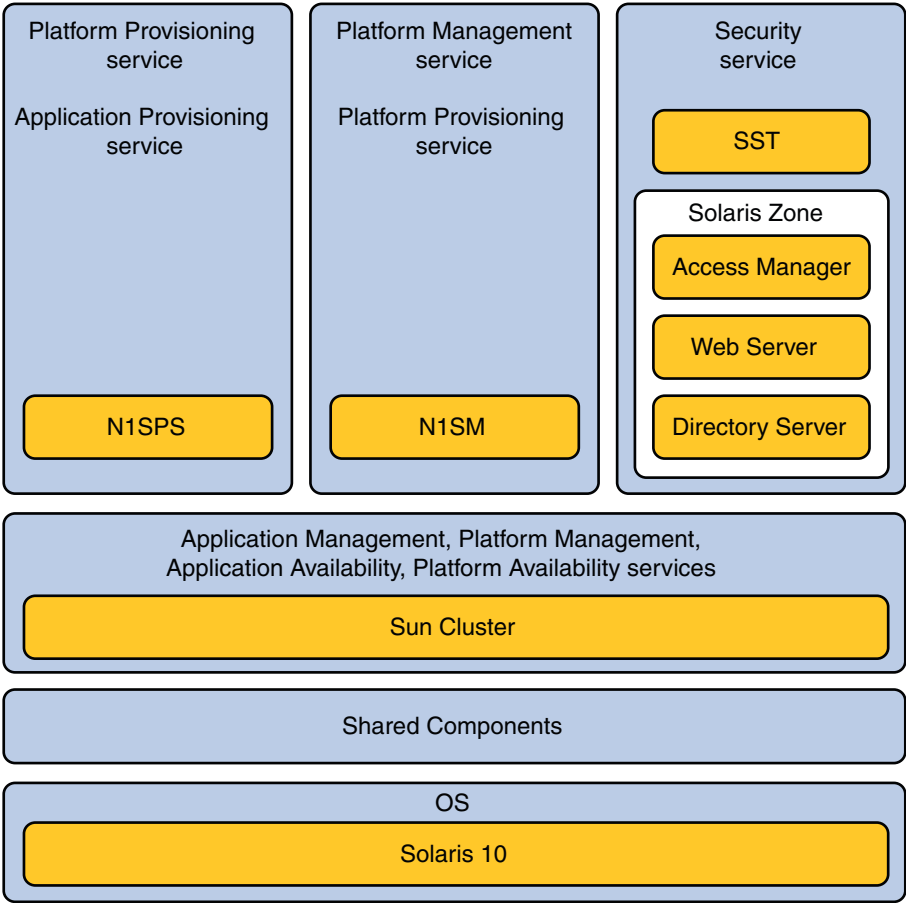


FIGURE P-1 Component Product Mapping

Supported Versions

The following table shows the OS and component versions that are supported by Sun OTP 2.0.

TABLE P-1 Sun OTP 2.0 Supported Versions

OS and Components	Version
Solaris™ OS	10 Update 3

TABLE P-1 Sun OTP 2.0 Supported Versions (Continued)

OS and Components	Version
Sun Cluster	3.2
Sun N1™ Service Provisioning System	5.2.4
Sun N1 System Manager	1.3.3
OS Provisioning Plug-in	3.2
Sun Java™ System Web Server	7.0 Update 1
Sun Java System Directory Server	6.1 Enterprise Edition
Sun Java System Access Manager	7.1
Solaris Security Toolkit	4.2

Sun OTP Documentation Set

Sun OTP guides are available as online files in PDF and HTML formats. The following table lists the Sun OTP guides and their purpose.

TABLE P-2 Sun OTP Documentation Set

Documentation	Purpose
<i>Sun Open Telecommunications Platform 2.0 Release Notes</i>	Late-breaking information about the software and documentation
<i>Sun Open Telecommunications Platform 2.0 Installation Guide</i>	To install Sun OTP in the development environment
<i>Sun Open Telecommunications Platform 2.0 Administration Guide</i>	To administer, upgrade, update, backup, and restore Sun OTP
<i>Sun Open Telecommunications Platform 2.0 Developer's Guide</i>	<ul style="list-style-type: none"> ■ To develop and deploy applications using Sun OTP ■ To describe the Sun OTP tools for developing applications ■ To adapt applications for Sun OTP ■ To install Sun OTP in the existing solution ■ To use Sun OTP for end-to-end solution deployment

The complete Sun OTP documentation is available at
<http://docs.sun.com/app/docs/coll/1629.4>.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the full document title and part number. The part number is located on the book's title page and in the document's URL. For example, the name of this book is Sun Open Telecommunications Platform 2.0 Developer's Guide and the part number of this book is 820-3065.

Supported Solution Topologies

You can deploy single-tier and multi-tier solutions using a centralized provisioning server of Sun OTP.

You can set up the provisioning server in the following ways:

- If you have a separate physical system for the provisioning server, set up the provisioning server on that system. This provisioning server is referred to as external provisioning server.
- If you do not have a separate physical system for the provisioning server, set up the provisioning server in the stand-alone host or in the first host of the cluster. This provisioning server is referred to as self-contained provisioning server.

External provisioning server is deployed on top of a separate Sun OTP instance. Self-contained provisioning server is deployed along with NEP applications on top of the Sun OTP instance and within the same operating system.

Solution Topologies

The following solution topologies are supported:

- [“One Provisioning Server for Each Sun OTP Instance” on page 13](#)
- [“One Provisioning Server for Each Tier” on page 14](#)
- [“One Provisioning Server for Each Solution Instance” on page 15](#)

One Provisioning Server for Each Sun OTP Instance

In this topology, one provisioning server is present for each Sun OTP instance.

In tier 3, the provisioning server within the normal box indicates the active node whereas the provisioning server within the dotted line box indicates the failover node or additional node. The figure also shows where you can place your applications in each tier.

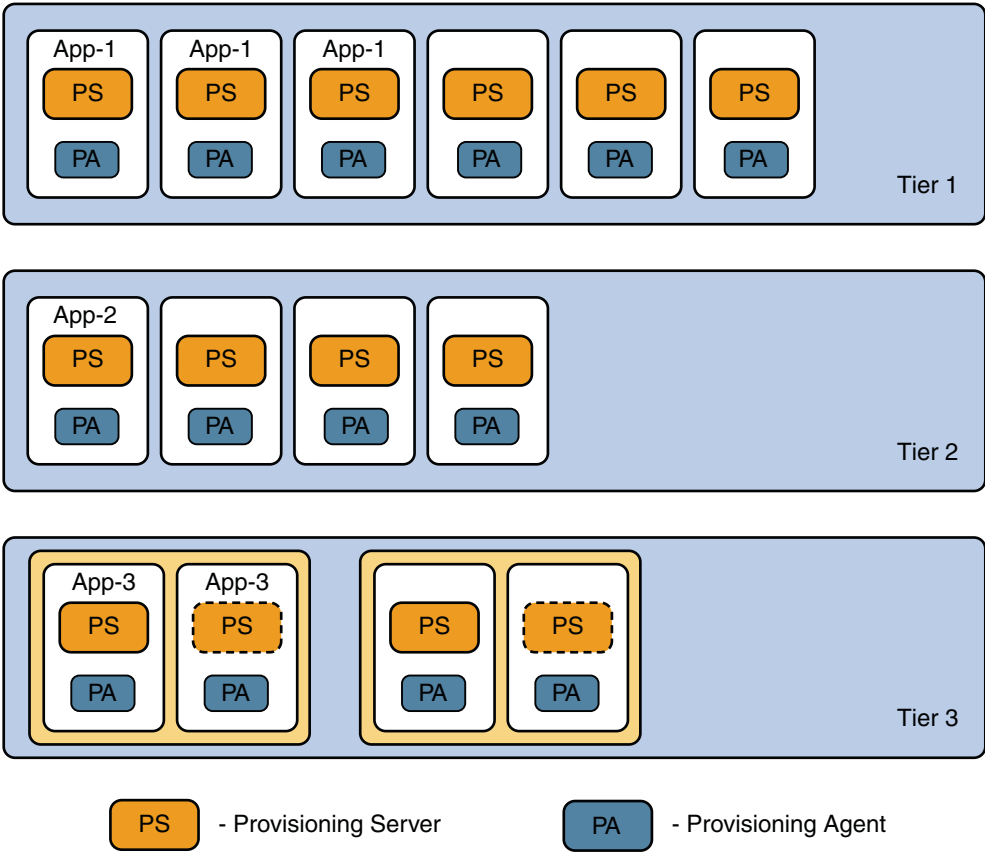


FIGURE 1-1 One Provisioning Server for Each Sun OTP Instance

One Provisioning Server for Each Tier

In this topology, one provisioning server is present for each tier.

In tier 3, the provisioning server within the normal box indicates the active node whereas the provisioning server within the dotted line box indicates the failover node or additional node. The figure also shows where you can place your applications in each tier.

In tier 3, the provisioning server is disabled in the second clustered OTP instance.

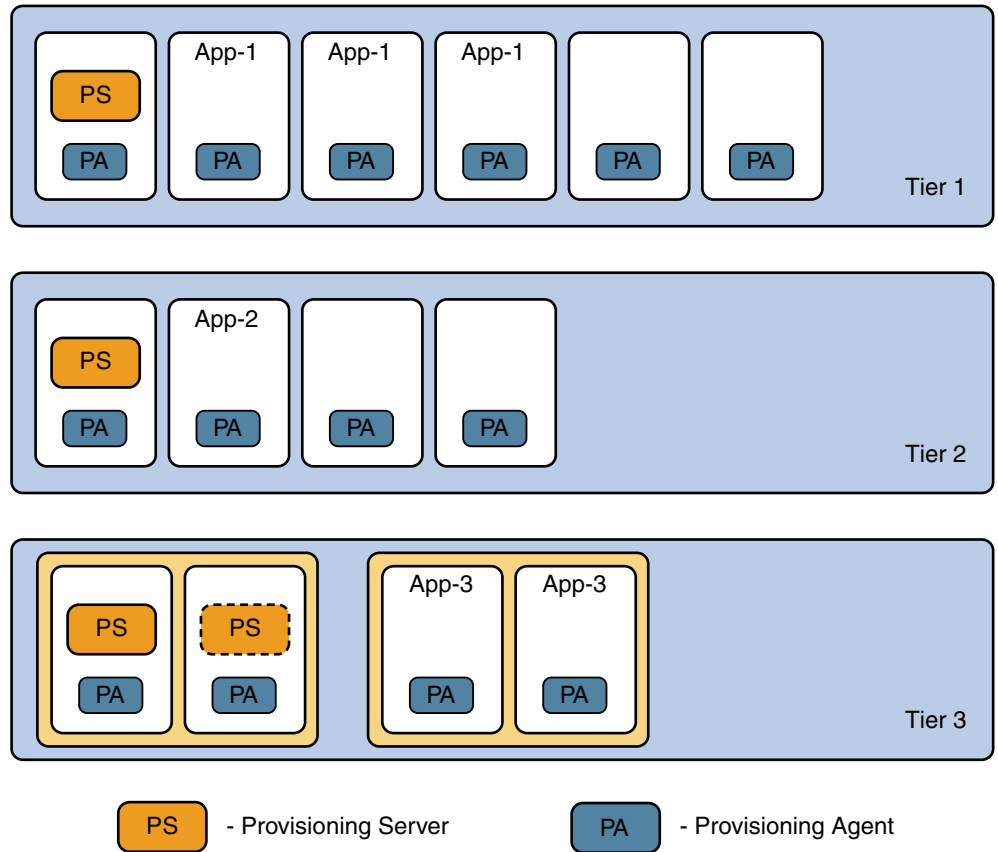


FIGURE 1-2 One Provisioning Server for Each Tier

One Provisioning Server for Each Solution Instance

In this topology, one provisioning server is present for the entire solution instance.

In tier 3, the provisioning server within the normal box indicates the active node whereas the provisioning server within the dotted line box indicates the failover node or additional node. The figure also shows where you can place your applications in each tier.

In tier 3, the provisioning server is disabled in the second clustered OTP instance.

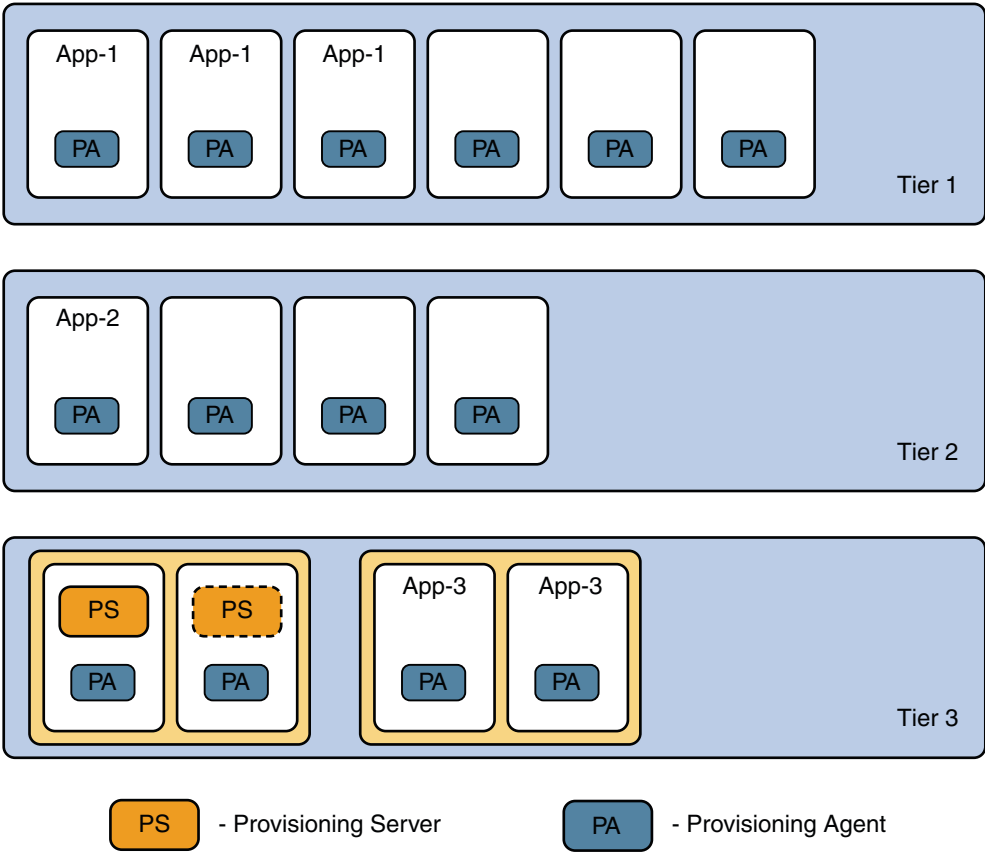


FIGURE 1-3 One Provisioning Server for Each Solution Instance

Developing Applications Using Sun OTP

This chapter lists the application programming interfaces (APIs) and protocols you can use for application development.

- “Interfaces Exposed by Sun OTP” on page 17
- “Sun OTP Protocols” on page 18

Interfaces Exposed by Sun OTP

The following table lists the interfaces exposed by Sun Open Telecommunications Platform.

TABLE 2-1 Sun OTP Interfaces

OS and Components	Interfaces	Documentation
Solaris™ 10 OS	<ul style="list-style-type: none"> ■ POSIX.1 (IEEE Std 1003.1) ■ POSIX.2 (IEEE Std 1003.2) ■ PAM (Pluggable Authentication Modules) 	POSIX.1(5) man page POSIX.1(5) POSIX.2(5) man page POSIX.2(5) <i>Solaris Security for Developers Guide</i> libpam (3LIB) man page libpam(3LIB)
Java™	<ul style="list-style-type: none"> ■ Java SE 1.4.2 Java interfaces ■ Java SE 5.0 Java interfaces ■ CORBA 	(http://java.sun.com/j2se/1.4.2/docs/) (http://java.sun.com/j2se/1.5.0/docs/) (http://java.sun.com/j2se/1.5.0/docs/guide/idl/)

TABLE 2-1 Sun OTP Interfaces (Continued)

OS and Components	Interfaces	Documentation
Sun N1™ Service Provisioning System 5.2.4	<ul style="list-style-type: none">■ XML■ Java APIs■ CLI■ GUI	<ul style="list-style-type: none">■ <i>Sun N1 Service Provisioning System 5.2 XML Schema Reference Guide</i>■ <i>Sun N1 Service Provisioning System JavaDoc</i>■ <i>Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual</i>■ <i>Sun N1 Service Provisioning System 5.2 System Administration Guide</i>
Sun N1 System Manager 1.3.3	<ul style="list-style-type: none">■ Shell scripts■ CLI■ GUI	<ul style="list-style-type: none">■ <i>Sun N1 System Manager 1.3 Command Line Reference Manual</i>■ Appendix A, “Sun N1 System Manager Protocol, Ports, and Features Reference,” in <i>Sun N1 System Manager 1.3 Installation and Configuration Guide</i>■ <i>Sun N1 System Manager 1.3 Discovery and Administration Guide</i>
Sun Cluster 3.2	<ul style="list-style-type: none">■ Sun Cluster Manager■ HA Agent Development Libraries■ CLI	<ul style="list-style-type: none">■ Appendix A, “Sun Cluster Object-Oriented Commands,” in <i>Sun Cluster System Administration Guide for Solaris OS</i>■ <i>Sun Cluster Data Services Developer's Guide for Solaris OS</i>

Sun OTP Protocols

The following table lists the protocols used by Sun Open Telecommunications Platform.

TABLE 2-2 Sun OTP Protocols

Protocols	Components	Documentation
TCP/IP	Solaris 10 OS	tcp(7P) and ip(7P)
SNMP (Net-snmp SNMP V3)	Solaris 10 OS	<i>Solaris System Management Agent Developer's Guide</i>
SSH/SSL	Solaris 10 OS	ssh(1) openssl(5)

TABLE 2-2 Sun OTP Protocols (Continued)

Protocols	Components	Documentation
DNS	Solaris 10 OS	<i>System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)</i>
iSCSI	Solaris 10 OS	<i>System Administration Guide: Devices and File Systems</i>
FC (FCP) ANSI X3.269-1996	Solaris 10 OS	<i>Solaris Fibre Channel Storage Configuration and Multipathing Support Guide</i>
RMI	Java 2 SDK SE 1.4.2 and Java SE 5.0 platform	(http://java.sun.com/j2se/1.4.2/docs/guide/rmi/) (http://java.sun.com/j2se/1.5.0/docs/guide/rmi/)
IIOP (RMI-IIOP)	Java 2 SDK SE 1.4.2 and Java SE 5.0 platform	(http://java.sun.com/j2se/1.4.2/docs/guide/rmi-iiop/) (http://java.sun.com/j2se/1.5.0/docs/guide/rmi-iiop/)

Sun OTP Tools for Application Development

This chapter lists the tools in Sun OTP that can be used for developing applications.

Tools for Application Development

TABLE 3-1 Sun OTP Tools for Application Development

Sun OTP Tools	Description	Documentation
Sun Cluster Agent Builder	Allows you to specify information about your application and the kind of resource type (scalable or failover) that you want to create.	<i>Sun Cluster Data Services Developer's Guide for Solaris OS</i>
Sun Cluster Generic Data Service	Allows your applications to become highly available or scalable by plugging them into the Sun Cluster Resource Group Management (RGM) framework.	<i>Sun Cluster Data Services Developer's Guide for Solaris OS</i>
Sun OTP Application Provisioning Service	Enterprise management software system that automates and accelerates the deployment of business services. It allows you to provision single-tier and multi-tier applications across heterogeneous environments.	<i>Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide</i> <i>Sun N1 Service Provisioning System 5.2 Operation and Provisioning Guide</i>

Sun OTP Components

This chapter describes Sun OTP components, plans, and plug-in. It also describes the Sun OTP deployment options and how you can customize the Sun OTP deployment.

- “N1 SPS Components” on page 23
- “N1 SPS Plans” on page 25
- “Sun OTP AHE Plug-in” on page 30
- “Customizing Sun OTP Deployment” on page 30

N1 SPS Components

The components installed by the N1 SPS (N1 Service Provisioning System) plans are as follows:

TABLE 4-1 N1 SPS Components

Components	Folder	Description
N1SPS	/com/sun/OTP/management/n1sps	Component to install Sun OTP application provisioning service (Sun N1 Service Provisioning System)
N1SM	/com/sun/OTP/management/n1sm	Component to install Sun OTP system management service (Sun N1 System Manager)
SunCluster	/com/sun/OTP/availability/sc	Component to install Sun OTP availability service (Sun Cluster)

TABLE 4-1 N1 SPS Components *(Continued)*

Components	Folder	Description
AccessManager	/com/sun/OTP/security/jes_am	Component to install Access Manager. This component is part of Sun OTP security service.
DirectoryServer	/com/sun/OTP/security/jes_ds	Component to install Directory Server. This component is part of Sun OTP security service.
WebServer	/com/sun/OTP/security/jes_ws	Component to install Web Server. This component is part of Sun OTP security service.
Zone, ZoneRASSH, ZoneRARAW	/com/sun/OTP/virtualization/zones	Components to install zones
sunotp-secure.driver.tar	/com/sun/OTP/security/jass	Component to administer Sun OTP JumpStart Architecture and Security Scripts (JASS)
PatchOS	/com/sun/OTP/Utilities	Component to install patches on the OS
SharedDisk	/com/sun/OTP/Utilities	Component to set up shared storage
WebSSO	/com/sun/OTP/security/WebSSO	Component to administer Sun OTP Web SSO
SharedComponents (internal to Sun OTP)	/com/sun/OTP/shared_components	Component to install Sun OTP shared services
JESHA (internal to Sun OTP)	/com/sun/OTP/security/jes_ha	Component to administer high availability services for Sun OTP security service
Init (internal to Sun OTP)	/com/sun/OTP/Utilities	Container Component that has constants and some common code. This common code is used by other components.
OTPConfig (internal to Sun OTP)	/com/sun/OTP/Utilities	Component to store and validate the Sun OTP configuration variables
SysMng (internal to Sun OTP)	/com/sun/OTP/N1SM	Component to administer N1SM

The following figure shows the N1 SPS folder structure of Sun OTP plug-in.

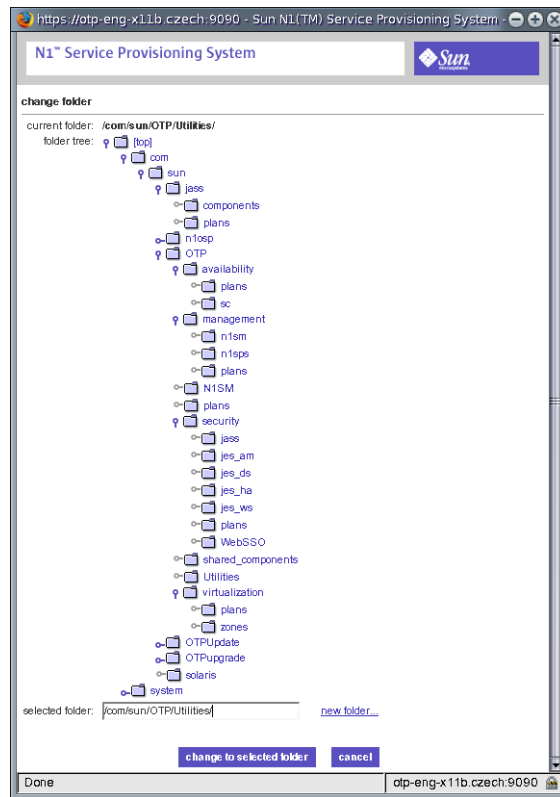


FIGURE 4-1 N1 SPS Folder Structure of Sun OTP Plug-in

N1 SPS Plans

This section describes the order of installing N1 SPS plans, plan re-runs, plan configuration files, and utility plans.

The following topics are discussed in this section:

- “Order of Installing N1 SPS Plans” on page 26
- “Plan Re-runs” on page 26
- “Plan Configuration Files” on page 27
- “Utility Plans” on page 30

Order of Installing N1 SPS Plans

A plan orchestrates deployment of multiple components.

To install Sun OTP, N1 SPS plans must be installed in the following order:

1. Setup Configuration
2. Patch
3. Install Availability Service
4. Configure Availability Service
5. Configure Solaris Containers (optional)
6. Setup Shared Storage (optional)
7. Install System Management Service
8. Install Provisioning Service
9. Install Security Service
10. Configure Components
11. Install Web SSO

Plan Re-runs

Plan re-runs provides an option to re-run specific plans instead of reinstalling Sun OTP from the beginning. Plans can be re-run multiple times irrespective of the previous state of the plan.

The following plans can be re-run:

- Set up Configuration
- Configure Solaris Containers
- Sun OTP system management service
- Sun OTP application provisioning service
- Sun OTP security service
- Configure Sun OTP services for high availability

When you re-run a plan, plan re-runs read the registry and determines the state of the plan. Based on the state of the plan, plan re-runs perform the following actions.

The following are the possible actions of plan re-runs:

- If you are running the plan for the first time, plan re-run runs the plan and updates the registry with the appropriate state.
- If you are re-running a successful plan, plan re-run completes without performing any installation.
- If you are re-running a failed plan, plan re-run uninstalls and reinstalls the components.

Plan Configuration Files

Plan configuration files are the variable sets provided by N1 SPS. You can create component variables for your plans using the variable sets. The Setup Configuration plan provided by Sun OTP accepts all the component variables and validates the variables provided in the variable set.

The following table lists the component variables used by Sun OTP plans. See [“Sun OTP Plan Settings Description” on page 81](#) for descriptions of each component variable.

TABLE 4-2 Component Variables Used by Sun OTP Plans

Plan	Component Variables
Patch	<ul style="list-style-type: none">mediaDirectoryapplyAllPatcheshostType
Install Availability Service	<ul style="list-style-type: none">mediaDirectoryhostTypeclusterNamesponsorNodeautoConfigureIPMPsecondaryInterfacesecondaryIPtestIPAddressprivateInterface1privateInterface2nodeAuthenticationtransportTypeInterface1transportTypeInterface2applyAllPatchesquorumAutoConfiguration

TABLE 4-2 Component Variables Used by Sun OTP Plans (Continued)

Plan	Component Variables
Configure Availability Service	<ul style="list-style-type: none">■ mediaDirectory■ hostType■ clusterName■ sponsorNode■ autoConfigureIPMP■ secondaryInterface■ secondaryIP■ testIPAddress■ privateInterface1■ privateInterface2■ nodeAuthentication■ transportTypeInterface1■ transportTypeInterface2■ applyAllPatches■ quorumAutoConfiguration
Configure Solaris Containers	<ul style="list-style-type: none">■ mediaDirectory■ hostType■ zoneName■ zoneInterface■ zoneIPAddress■ zonePath■ zoneHostname■ zoneMask■ zoneDefaultRoute■ zoneSPSHost
Setup Shared Storage	<ul style="list-style-type: none">■ rootDisk■ diskSlice■ hostType
Install System Management Service	<ul style="list-style-type: none">■ mediaDirectory■ hostType■ mgmtHost■ mgmtIP■ managementInterface■ provisioningInterface■ applyAllPatches

TABLE 4-2 Component Variables Used by Sun OTP Plans (Continued)

Plan	Component Variables
Install Provisioning Service	<ul style="list-style-type: none"> ■ mediaDirectory ■ mgmtHost ■ mgmtIP ■ applyAllPatches ■ spsRAConnectionType ■ hostType
Install Security Service	<ul style="list-style-type: none"> ■ mediaDirectory ■ hostType ■ hostName ■ zoneHostname ■ zoneIPAddress ■ jesHAHost ■ domainName ■ applyAllPatches ■ mmrHostList
Configure Components	<ul style="list-style-type: none"> ■ mediaDirectory ■ mgmtHost ■ mgmtIP ■ applyAllPatches ■ spsRAConnectionType ■ hostType ■ hostName ■ zoneHostname ■ zoneIPAddress ■ jesHAHost ■ jesHANodeList ■ domainName ■ mmrHostList
Install Web SSO	<ul style="list-style-type: none"> ■ hostName ■ hostType ■ mgmtHost ■ jesHAHost ■ ssoCookieDomain ■ zoneIPAddress ■ zoneHostname ■ domainName

Utility Plans

Sun OTP provides the following utility plans:

TABLE 4-3 Sun OTP Utility Plans

Plan	Description
Patch	Installs the OS patches on the Sun OTP host.
Convert	Converts the stand-alone Sun OTP host to the clustered Sun OTP host.
Enable & Disable	Enables and disables Sun OTP system management service and Sun OTP application provisioning service.
Add User	Adds Web SSO user.
Change User Password	Changes the password of existing Web SSO user.
Remove User	Removes Web SSO user.
Install Driver	Installs the Sun OTP SST package.
UnInstall Driver	Uninstalls the Sun OTP SST package.
Harden	Improves the security of the system by hardening the Sun OTP host. You can close the ports and disable the services that might present a security risk to the system.
UnHarden & Keep	Reopens the ports and enables the services that were closed by the hardening process. This plan rolls back only the unchanged configuration files to its default state.
UnHarden & Revert	Reopens the ports and enables the services that were closed by the hardening process. This plan rolls back all the configuration files, including the changed files, to its default state.

Sun OTP AHE Plug-in

A plug-in is a collection of plans and components. Sun OTP AHE plug-in is a collection of Sun OTP plans and components.

Customizing Sun OTP Deployment

Sun OTP plans are modularized in a way that NEP developers can insert their own customization. NEP developers can insert their customization before or after the Sun OTP plans. NEP developers can copy the Sun OTP AHE plug-in and customize the OTP deployment with native command execution or with new plans. They can also write their own variable set as part of the new plans.

Sun OTP components are meant to be called only from Sun OTP plans. Sun OTP plans can be called from custom plans only if the order of installation of Sun OTP plans described in [“Order of Installing N1 SPS Plans” on page 26](#) is satisfied.

Note – Do not customize the Sun OTP plans and Sun OTP variable set.

Examples of Solution Specific Steps

The following table lists some of the examples of solution specific steps and where it can be inserted in Sun OTP deployment.

TABLE 4-4 Examples of Solution Specific Steps

Examples of Solution Specific Steps	Insertion Point in Sun OTP Deployment
User Administration	After Sun OTP is installed
Quorum Configuration	After Sun OTP availability service is installed and configured
Naming Services	After OS is installed
Shared Storage Configuration	After Sun OTP availability service is installed and configured

Adapting Applications for Sun OTP

You need to create the provisioning and management plug-ins to adapt your application to Sun OTP. A plug-in is a collection of plans and components. A plan orchestrates deployment of multiple components.

The following topics are discussed:

- [“Developing N1 SPS Plans for NEP Application Provisioning” on page 33](#)
- [“Developing Sun Cluster HA Agents for NEP Application Management” on page 38](#)

Developing N1 SPS Plans for NEP Application Provisioning

Refer to *Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide* to create components and plans for NEP applications.

The three options of adapting your application for application provisioning are as follows:

[“Calling Existing Application Installers from N1 SPS” on page 33](#)

[“Moving an Existing Application Installer Code to an SPS Component” on page 34](#)

[“Using Native N1 SPS Provisioning Features” on page 36](#)

Calling Existing Application Installers from N1 SPS

If the installation script for your application already exists, you can create a component template and call the installation script from the component template.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<component xmlns='http://www.sun.com/schema/SPS'
name='DeployMyComponent'
```

```
version='5.0'
description='Deploy and configure My Component'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
path='/com/sun/foobar'
author='OEM Platforms'
softwareVendor='Sun Microsystems'
xsi:schemaLocation='http://www.sun.com/schema/SPS component.xsd'

<extends>
<type name='system#container'></type>
</extends>

<varList>
  <var name="bashShell" modifier="FINAL" default="/usr/bin/bash"/>
  <var name="installPath" default="/opt"/>
</varList>

<installList>
  <installSteps name="deployMyComponent" access="PUBLIC">

    <execNative userToRunAs='root' timeout='7200'>
      <inputText><![CDATA[

        #run installer
        /script_directory/installer.sh -p param1 -s param2
        #check for errors
        RET=$?; if [ $RET != 0 ]; then exit $RET; fi

      ]]></inputText> <exec cmd=":[bashShell]"></exec>
    </execNative>
  </installSteps>
</installList>
</component>
```

Moving an Existing Application Installer Code to an SPS Component

You can also copy the application deployment logic from the installation script of your application and paste the logic into a provisioning component. This option can be used for a tighter integration with Sun OTP provisioning service. In the following example, the application installation script is embedded in its provisioning component.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<component xmlns='http://www.sun.com/schema/SPS' name='InstallMyComponent'
  version='5.2.4'
```

```

description='Install My Component'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
author='OEM Platforms' softwareVendor='Sun Microsystems'
path='/com/sun/examples'
xsi:schemaLocation='http://www.sun.com/schema/SPS component.xsd'

<extends>
<type name='system#container'></type>
</extends>

<varList>
<var name='shell' modifier='FINAL' default='/usr/bin/bash'></var>
<var name='installPath' default='/opt' modifier='FINAL'></var>
<var name="mediaDirectory" default="[component:/com/sun/example:mediaDirectory]"/>
<var name='postcheck' default='no' prompt="Mark yes if post check required"></var>
</varList>
<installList>
  <installSteps returns='false' name='InstallMyComponent' requireLocking='true'>
    <execNative userToRunAs='root' timeout='7200'>
      <inputText><![CDATA[

# Example of product installation:
# -step 1: initial checks (does product bundle exist?)
# -step 2: extraction of bundle into temporary directory
# -step 3: invocation of product installer
# -step 4: (optional) installation post check (was product installed fine?)
# -step 5: temporary files clean up

# 1: initial checks
scriptPath=[mediaDirectory]/productName
if [ ! -d $scriptPath ]; then
echo "Error: $scriptPath does not exist"
exit 1
fi
if [ ! -f ${scriptPath}/productName.jar ]; then
echo "Error: productName.jar does not exist"
exit 1
fi

# 2a: copy bundle to tmp
mkdir -p /tmp/install/productName
cp ${scriptPath}/productName.jar /tmp/install/productName
cd /tmp/install/productName

# 2b: extraction of bundle contents
jar xf productName.jar
rm -f productName.jar

```

```
# 3: invoke installer
java -cp /tmp/install/productName com.company.product.Boot -p param1 -d param2 -i :[installPath]
RET=$?; if [ $RET != 0 ]; then exit $RET; fi

# 4: post check
if [ "yes" = :[postcheck] ]; then
out='java -cp /tmp/install/productName com.company.product.diagnostics checkInstallation -p param1 \
-d param2 -i :[installPath] | grep "Status=OK"'
RET=$?; if [ $RET != 0 ]; then exit $RET; fi
fi

# 5: cleanup
cd /
rm -rf /tmp/install/productName

]]></inputText>
<exec cmd=':[shell] '></exec>
</execNative>
</installSteps>
</installList>
</component>
```

The embedded code can be any interpreted shell code or Java code. The shell code can be specified by specifying the value for the `cmd` attribute in the `exec` element. The embedded Java code can be specified by specifying the `execJava` element.

Using Native N1 SPS Provisioning Features

To fully benefit from Sun OTP application provisioning service features, the target application deployment can be redesigned using native Sun OTP provisioning components. This can be done by mapping the application deployment elements to native provisioning components. Based on your application distribution model, these elements could be packages (Solaris SVR4, Linux RPM and Deb) or archives (such as zip, tar, jar, ear) or even individual files (like a library, a kernel module, or scripts).

The following sample code demonstrates package-based installation.

Example 1

```
<installList>
  <installSteps returns='false' name='default' requireLocking='true'>
    <execNative timeout='1800'>
      <inputText><![CDATA[>
        echo "Executing Pre-Install procedures before resource deployment"
        echo "This will be installed into: :[installPath]"
```

```

    ]]></inputText>
        <exec cmd='/bin/sh'></exec>
    </execNative>
    <deployResource></deployResource>
    <execNative timeout='1800'>
        <inputText><![CDATA[
            echo "Executing Post-Install procedures after resource deployment"
            pkgadd -d :[installPath] SampleAppPkg
        ]]></inputText>
        <exec cmd='/bin/sh'></exec>
    </execNative>
</installSteps>
</installList>

```

The following sample code demonstrates package-based uninstallation.

Example 2

```

<uninstallList>
    <uninstallSteps returns='false' name='default' requireLocking='true'>
        <execNative timeout='1800'>
            <inputText><![CDATA[
                echo "Executing Pre-Uninstall procedures before resource undeployment"
            ]]></inputText>
            <exec cmd='/bin/sh'></exec>
        </execNative>
        <undeployResource></undeployResource>
        <execNative timeout='1800'>
            <inputText><![CDATA[
                echo "Executing Post-Uninstall procedures after resource undeployment"
                yes | pkgrm SampleAppPkg
            ]]></inputText>
            <exec cmd='/bin/sh'></exec>
        </execNative>
    </uninstallSteps>
</uninstallList>

```

Refer to the following guides to create plans, components, and plug-ins using N1 SPS.

- *Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide*
- *Sun N1 Service Provisioning System 5.2 Operation and Provisioning Guide*
- *Sun N1 Service Provisioning System 5.2 Plug-in Development Guide*
- [N1 Service Provisioning System 5.2.1 Plug-In Collection](#)
- *Best Practices Walkthrough* document at <http://www.sun.com/bigadmin/hubs/sysmgmt/learn/training.jsp>
- *Developer Guidelines* document at <http://www.sun.com/bigadmin/hubs/sysmgmt/learn/training.jsp>

Developing Sun Cluster HA Agents for NEP Application Management

You need to create the management agents for your application to use the Sun OTP availability service. You can develop agents using agent builder, scripts, and C/C++.

Developing Agents Using Agent Builder

Sun Cluster agent builder is included in Sun OTP. The agent builder allows you to specify information about your application and the kind of resource type (scalable or failover) that you want to create.

You need to provide start, stop (optional), validate (optional), and probe (optional) commands for your application. Agent builder automatically generates an agent package that can be used to put your application under Sun OTP management service control.

For more information on developing agents using agent builder, refer to *Sun Cluster Data Services Developer's Guide for Solaris OS*.

Developing Agents Using Scripts and C/C++

Refer to the following guides to develop agents using scripts and C/C++.

- *Sun Cluster Data Services Developer's Guide for Solaris OS*
- [Sun Cluster 3.2 Data Services Collection for Solaris OS \(SPARC Platform Edition\)](#)
- *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*

Installing Sun OTP As Part of Existing Solution Deployment

This chapter describes how to invoke the deployment plans from existing installation environments.

The following topics are discussed:

- “Calling N1 SPS Plans From Shell Scripts” on page 39
- “Calling N1 SPS Plans From C/C++” on page 41
- “Calling N1 SPS Plans From Java” on page 43
- “Using N1 SPS From JumpStart” on page 45
- “Using N1 SPS From JET” on page 46

Calling N1 SPS Plans From Shell Scripts

You can invoke the N1 SPS plans through the Command-Line Interface (CLI) by using the `cr_cli` binary. On a host that has Sun OTP 2.0, `cr_cli` is located in the `/var/js/spsothp/N1_Service_Provisioning_System_5.2/cli/bin/cr_cli` directory. For detailed description of the N1 SPS CLI, see *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

A typical workflow of invoking the N1 SPS plan from a script consists of three steps. The script performs the following tasks:

1. Creates a variable setting for the components that are installed as part of the plan. Alternatively, you can also use the default values of variables for a component.
2. Invokes the `pe.p.run` command through the N1 SPS CLI, which runs the specified plan with either the default values for variables or using specified variable settings.

Note – The choice between the default values and variable settings is done for each component that is installed as part of the plan.

3. Waits until the running plan is completed and checks for errors.

Examples

An example for creating the variable settings for component foo is as follows:

```
/var/js/spsotp/N1_Service_Provisioning_System_5.2/cli/bin/cr_cli -cmd
cdb.vs.add -u otpadmin -p adminadmin -comp NM:/path/to/component/foo -name
foo-varset1 -vars "var1=value1;var2=value2"
```

The above example creates a variable setting for the component foo with the name foo-varset1. You can use the foo-varset1 variable in the vs parameter of the pe.p.run command for plans that execute an install block on component foo.

An example for running the InstallSecurity plan from Sun OTP 2.0 plug-in is as follows:

```
/var/js/spsotp/N1_Service_Provisioning_System_5.2/cli/bin/cr_cli -cmd pe.p.run
-u otpadmin -p adminadmin -PID NM:/com/sun/OTP/security/plans/InstallSecurity
-tar "H:NM:otp-eng-x11;H:NM:otp-eng-x11;H:NM:otp-eng-x11" -comp
"+;++;+,+,+,+,+,+,+" -vs "+;++;+,+,+,+,+,+,+" -pto 9000 -nto 9000
```

In the above example, the InstallSecurity plan is run on a host named host otp-eng-x11. The user name is otpadmin and password is adminadmin. The default values for component variables are used and the time out is set to 9000 seconds.

Note the syntax of the vs parameter specifying the variable settings to use. The InstallSecurity plan consists of three sub plans. Each sub plan takes a part of the vs parameter value and delimits it by semicolons, that is, the first sub plan takes +, the second takes + and the third takes +, +, +, +, +, +, +, +. The third sub plan calls eight install blocks on various components. For each of these blocks, the default variable values are being used, which is specified by the 8 (+) plus signs and (,) comma as a delimiter. Similar rules apply to the comp parameter. The tar parameter specifying the target host consists of three parts that specify the target for each of the three sub plans.

For detailed information about the pe.p.run command, see *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

If you use the correct syntax, the command will complete successfully and return a ID. You can use this ID to check the plan status by running the pe.p.lo command. Error checking needs to parse the pe.p.lo command output.

The following example waits for running task ID
010018158098-1200908797135-01929-1300304153.

```
/var/js/spsotp/N1_Service_Provisioning_System_5.2/cli/bin/cr_cli -cmd pe.p.lo
-u otpadmin -p adminadmin -ID 010018158098-1200908797135-01929-1300304153
```


The output of this example lists the sub-tasks that run in the background before completing this task.

Calling N1 SPS Plans From C/C++

Since there is no N1 SPS C/C++ interface available, you can invoke the plans using N1 SPS CLI. Two N1 SPS CLI invocations are required.

- The code runs the plan first. The N1 SPS CLI returns the ID of the newly created running task on a standard output.
- To wait for task completion and check on the task progress, the code runs the N1 SPS CLI command with the `-cmd pe.p.lo` parameter and the ID of the task from the previous step.

For detailed information about N1 SPS CLI, see *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

The following sample code illustrates how to run a plan from the C++ code.

```
/*
 * Example of running an N1SPS plan from C++ code.
 *
 * To simplify this example, not all possible errors have been handled
 * properly.
 */

////////////////////////////////////

#include<iostream>
#include<sstream>
#include<cstdlib>

////////////////////////////////////

//path to N1SPS CLI
#define N1SPS_CLI "/var/js/spsotp/N1_Service_Provisioning_System_5.2/cli/bin/cr_cli"

//plan to run
#define PLAN "/com/sun/OTP/security/plans/InstallSecurity"

//variable setting version string - see N1SPS CLI doc for more info
#define PLAN_COMP_DEFAULT_VS "+;+;+,+,+,+,+,+,+,+"

//component version string - N1SPS CLI doc for more info
#define DEFAULT_COMP_DEFAULT_VS

//timeout in seconds - see N1SPS CLI doc (pto and nto parameters) for details
```

```

#define TIMEOUT_SEC 600

#define BUFF_SIZE 100

////////////////////////////////////////////////////////////////

using namespace std;

/**
 * prints usage message
 */
void usage() {
    cerr << "Usage:" << endl
         << "listTasks <login> <password> <target>" << endl;
}

////////////////////////////////////////////////////////////////

/**
 * main function
 * expected argv content: login, password, target
 */
int main(int argc, char **argv) {

    if( argc != 4 ) {
        //incorrect number of parameters, so print usage information
        usage();
    } else {
        cout << "Running plan " << PLAN << ":" << endl << endl;

        //prepare string for invocation of N1SPS CLI to run a plan
        stringstream s;
        s << N1SPS_CLI << " -cmd pe.p.run -u " << argv[1] << " -p " << argv[2]
          << " -tar H:NM:" << argv[3] << " -PID NM:" << PLAN
          << " -comp " << DEFAULT_COMP << " -vs " << DEFAULT_VS
          << " -pto " << TIMEOUT_SEC << " -nto " << TIMEOUT_SEC;

        FILE *f = NULL;
        if( (f = popen(s.str().c_str(), "r")) == NULL ) {
            //popen failed (fork() or pipe() failed), so print error message
            cerr << "Error running plan " << PLAN << ": popen() failed" << endl;
        } else {
            //read N1SPS CLI output (i.e. Task ID of running plan)
            char buff[BUFF_SIZE];
            fgets(buff, BUFF_SIZE, f);

            //wait for N1SPS CLI to finish
            if( pclose(f) != 0 ) {

```

```

        cerr << "Error running plan " << PLAN << ": N1SPS CLI failed" << endl;
        return -1;
    }

    cout << "Plan " << PLAN << "execution started." << endl
        << "Task ID:" << buff << endl;

    /*
     * prepare a new string for N1SPS CLI invocation to check on progress
     * of the running task
     */
    s.str("");
    s << N1SPS_CLI << " -cmd pe.p.lo -u " << argv[1] << " -p " << argv[2]
        << " -ID " << buff;

    cout << "Checking progress of task ID " << buff << endl;

    /*
     * wait for the running task to finish
     *
     * return the same return code N1SPS CLI returns
     */
    return system(s.str().c_str());
}
}
return -1;
}

```

To get a code in plain C, replace all stream related C++ specific code to equivalent code in C.

Calling N1 SPS Plans From Java

You can invoke the N1 SPS plans through the Java API by using the `sps-api.jar` library. This library is located in the `/var/js/spsotp/N1_Service_Provisioning_System_5.2/cli/lib/` directory. For more information about the API documentation, see *Sun N1 Service Provisioning System JavaDoc*.

A sample code indicating the running of the `InstallSecurity` plan from Sun OTP 2.0 plug-in and waiting for running plan of given ID is as follows:

Note – This example does not handle all plan execution scenarios. Consider a case where there are 20 tasks that should be completed after the task that you specify completes. Before you check the status of task, the array that the `pe.p.la` command generates does not contain any information regarding the task. However, in case the array does not contain any information corresponding to your task, you can rerun the `pe.p.la` command specifying a higher result array size or by specifying a plan name for the tasks to match.

For more information about the `pe.p.la` and `pe.p.lo` commands, see *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

```
package spsapitest;

import com.sun.n1.sps.client.*;
import com.sun.n1.sps.model.executor.RunningPlanBean;
import com.sun.n1.sps.model.executor.TaskID;
import com.sun.n1.sps.model.util.ClientException;
import java.io.File;
import java.util.HashMap;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;

public class SPSTest {

    public static void main(String[] args) {
        try {
            CommandManagerBuilder cmb = new com.sun.n1.sps.client.CommandManagerBuilder();
            cmb.setCLIInstallationDir(new File("/var/js/spsothp/N1_Service_Provisioning_System_5.2/cli"));
            CommandManager cm = cmb.build();

            //running the plan
            Map<String, String> arguments = new HashMap<String, String>();
            arguments.put("u", "otpadmin"); //user
            arguments.put("p", "adminadmin"); //password
            arguments.put("PID", "NM:/com/sun/OTP/security/plans/InstallSecurity"); //plan name
            arguments.put("tar", "H:Nm:otp-eng-x11;H:Nm:otp-eng-x11;H:Nm:otp-eng-x11"); //target
            arguments.put("vs", "+;+;+;+;+;+;+;+;+;+"); //variable settings
            arguments.put("comp", "+;+;+;+;+;+;+;+;+;+"); //components versions
            arguments.put("pto", "9000"); //plan timeout
            arguments.put("nto", "9000"); //native call timeout
            TaskID taskID = (TaskID)cm.execute("pe.p.run", arguments); //see pe.p.run in N1SPS CLI doc

            //waiting for plan to complete
            arguments.clear();
            arguments.put("u", "otpadmin");
```

```

arguments.put("p", "adminadmin");
arguments.put("ID", taskID.toString());
cm.execute("pe.p.lo", arguments);

//checking plan status
arguments.clear();
arguments.put("u", "otpadmin");
arguments.put("p", "adminadmin");
arguments.put("max", "20");           //maximum numbers of tasks to get info
RunningPlanBean[] rpba = (RunningPlanBean[]) cm.execute("pe.p.la", arguments);
for (RunningPlanBean rpb : rpba) {
    if (taskID.equals(rpb.getTaskID())) {
        System.out.println("Task ID: "+rpb.getTaskID().toString()+"\n"+
            "Plan name: "+rpb.getName()+"\n"+
            "Status: "+rpb.getTaskStatus().toString()+"\n"+
            "Start date: "+rpb.getStartDate().toString()+"\n"+
            "Complete date: "+rpb.getCompleteDate());
    }
}
} catch (ConfigurationException ex) {
    Logger.getLogger(SPSTest.class.getName()).log(Level.SEVERE, null, ex);
} catch (ClientException ex) {
    Logger.getLogger(SPSTest.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

Using N1 SPS From JumpStart

To invoke N1 SPS through JumpStart, perform the following steps.

1. Place the flash archive binary into the right location. The flash archive for Sun OTP is included in the bootable DVD that can be copied to jump start the system.
2. Create the JumpStart profile for flash archive.

```

# cat profile-flash
install_type          flash_install
archive_location nfs  <flash_archive_location>
partitioning          explicit
filesystems           c0t0d0s0 free /
filesystems           c0t0d0s1 4096 swap
filesystems           c0t0d0s3 512 /globaldevices
filesystems           c0t0d0s7 128

```

3. Modify the rule file to point to the profile you created.

```
hostname otpclient37    sparcsparc -          profile-flash
```

4. Update the JumpStart rules file by running the `~/rules_directory/check` command.
5. Update the JumpStart server using the `addclient` command to enable booting of the system.
6. Log in to the system console and type:
`ok boot net - install`
7. After the OS is installed, set up the Sun OTP Provisioning Service. See “To Install Sun OTP Using Sun OTP Bootable DVD Image” in *Sun Open Telecommunications Platform 2.0 Installation Guide*.
8. Deploy the Sun OTP AHE environment through the Sun OTP Provisioning Service. See “To Install Sun OTP Using Sun OTP Bootable DVD Image” in *Sun Open Telecommunications Platform 2.0 Installation Guide*.

A sample JumpStart profile for a flash archive is as follows:

sample jumpstart profile for flash archive

```
samplel@sr1-usca-22:/net/otpininstall/export/jumpstart# cat profile-flash
install_type          *flash_install*
archive_location nfs  *anew20:/export/Flashes/sample_otp_sun4u*
partitioning          explicit
fileys                c0t0d0s0 free /
fileys                c0t0d0s1 4096 swap
fileys                c0t0d0s3 512 /globaldevices
fileys                c0t0d0s7 128
```

Modify the rules files to point to archive.

```
hostname otpclient37   sparc -           profile-flash
```

Using N1 SPS From JET

You can add additional JET modules to the Solaris boot and install (JET) server. Although the process is described in the context of a flash module, the logic extends to other JET modules. For instructions on how to add a flash module, see “How to Add a Flash Module” in *Sun N1 Service Provisioning System User’s Guide for OS Provisioning Plug-In 3.1*.

Using Sun OTP for End-to-End Solution Deployment

This chapter lists the stages in using Sun OTP for end-to-end solution deployment.

Stages in Using Sun OTP for End-to-End Solution Deployment

Before You Begin: Before using Sun OTP for end-to-end solution deployment, you need to create N1 SPS solution plans. The N1 SPS solution plans consist of Sun OTP plans and NEP specific plans. The N1 SPS solution plans allows you to deploy OS, Sun OTP components, and NEP applications.

After creating N1 SPS solution plans, you typically follow these stages in this order for end-to-end solution deployment.

1. Deploy operating systems from solution plans.
2. Deploy Sun OTP components from solution plans.
3. Deploy hosted NEP applications from solution plans. NEP applications can also be installed between the Sun OTP plans.

The following figure shows the deployment path and deployment tools available in Sun OTP.

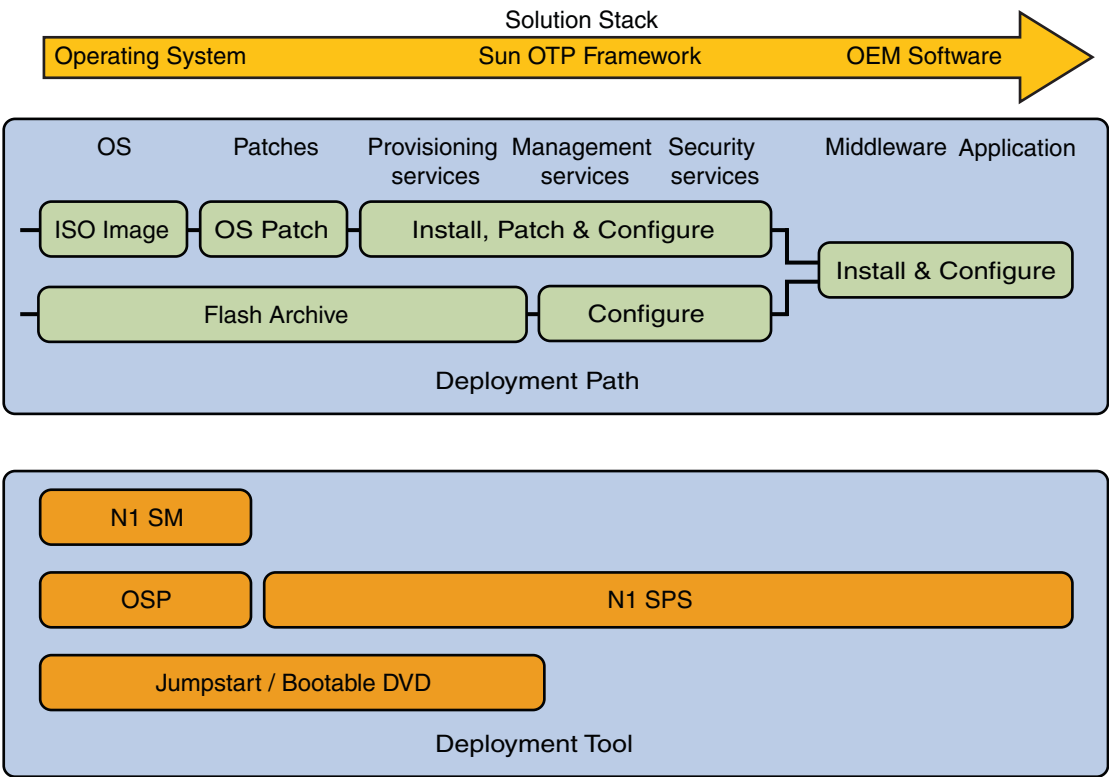


FIGURE 7-1 Sun OTP in End-to-End Solution Deployment

The following figure shows several control points that are available in Sun OTP solution deployment. These control points indicate that NEP specific plans can be inserted at these points. However, the order of installation of Sun OTP plans described in [“Order of Installing N1 SPS Plans” on page 26](#) must be satisfied.

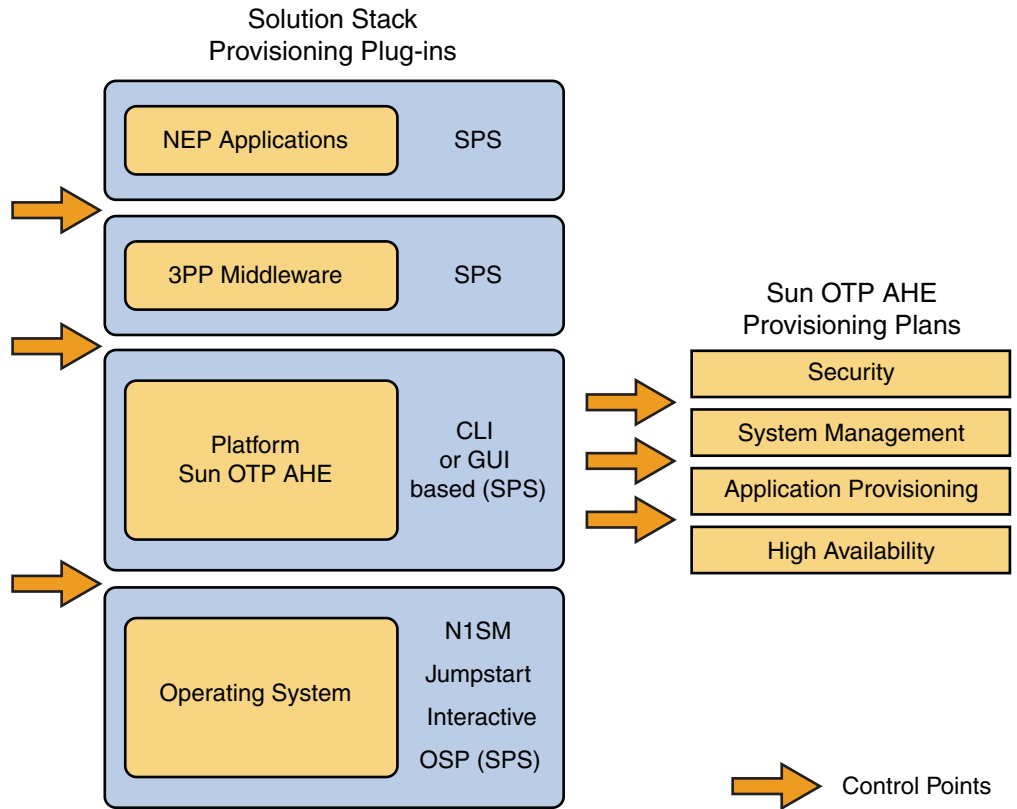


FIGURE 7-2 Control Points Implementation

Deployment Methods

The following table lists the deployment methods that you can use for end-to-end solution deployment.

TABLE 7-1 Deployment Methods

Component	Method
Solaris OS	<ul style="list-style-type: none"> JumpStart or JET Bootable Sun OTP DVD N1 SM GUI/CLI N1 SPS GUI/CLI/JAVA API

TABLE 7-1 Deployment Methods (Continued)

Component	Method
Sun OTP	<ul style="list-style-type: none">■ JumpStart or JET■ Bootable Sun OTP DVD■ N1 SPS GUI/CLI/JAVA API
Third-party products	N1 SPS GUI/CLI/JAVA API
NEP applications	N1 SPS GUI/CLI/JAVA API

Using Sun OTP Security Services

Sun OTP 2.0 includes Access Manager, Web Server, and Directory Server. Do not deploy any additional web applications under the web server that is co-packaged with Sun OTP 2.0. If you require to deploy additional web applications, deploy it under a different web server.

The following topics are discussed:

- “Securing Web Applications With Sun OTP” on page 51
- “Integrating Web Applications Into Sun OTP Web SSO Environment” on page 54
- “Hardening Sun OTP and Hosted NEP Applications” on page 56

Securing Web Applications With Sun OTP

This section describes the various methods to secure web applications with Sun OTP. You can configure the web server to use the Policy Agent software. The Policy Agent software helps in restricting access to the server web pages.

Configuring Policy Agent

The standard way to secure a web application is to install Access Manager's Policy Agent on a web server or an application server, and the agent protects the defined resources on the container. It is necessary to change the configuration file for the Policy Agent. Change the username and password that the agent uses to connect to the Access Manager. For Web Server 6.1 on Solaris, see “Updating the Agent Profile Name and the Agent Profile Password in Web Agents” in *Sun Java System Access Manager Policy Agent 2.2 Guide for Sun Java System Web Server 6.1*.

For more information, see *Sun Java System Access Manager Policy Agent 2.2 User's Guide* and Sun Java System Access Manager 7.1 documentation collection at <http://docs.sun.com/app/docs/coll/1292.2>.

▼ To Configure Policy Agent for Web Server 6.1 on Solaris 10 11/06

- 1 **Go to the config directory.**
`cd /etc/opt/SUNWam/agents/es6/config/...instance.../`
- 2 **Edit the `AMAgent.properties` file.**
- 3 **Create a user name that you will use to log in to Access Manager. For example,**
`com.sun.am.policy.am.username = amAdmin.`
- 4 **Check the settings `com.sun.am.naming.url` and `com.sun.am.policy.am.login.url`.**
 The settings should point to the URLs where the naming service and login pages of Access Manager are available.
- 5 **Go to the bin directory.**
`cd /opt/SUNWam/agents/bin`
- 6 **Encrypt the password for the user by using the `crypt_util` utility. For example, `crypt_util secretPassword`.**

Note – The `crypt_util` utility is part of the Access Manager installation.

- 7 **Edit the `AMAgent.properties` file again to set a password for the user name that you created in Step 3. Use the hash generated by `crypt_util`. For example,**
`com.sun.am.policy.am.password = XFC3z18nqMEgWbnshtNfwQ==`
 For more information, see *Sun Java System Access Manager Policy Agent 2.2 User's Guide* and [Sun Java System Access Manager 7.1 Documentation Collection](#).

Example of a Protected JSP Page

The following code illustrates a JavaServer Page (JSP) with the login and logout functionality. This page has a section that can be viewed only by authenticated users. This code uses the Single Sign On interface of Access Manager.

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@page import="com.iplanet.sso.*"%>
<h1>AuthTest</h1>

<p>
  <a href="http://ipp-node11.czech.sun.com/amserver/UI/Login?org=dc%3Dczech%2C
dc%3Dsun%2Cdc%3Dcom&goto=/test.jsp&gotoOnFail=/error.jsp">Login</a>
|
```

```

    <a href="http://ipp-node11.czech.sun.com/amserver/UI/Logout?
goto=/test.jsp">Logout</a>
</p>

<%
SSOTokenManager ssoManager = SSOTokenManager.getInstance();
SSOToken ssoToken = null;
String orgName = "";

boolean ok = false;
try{
    ssoToken = ssoManager.createSSOToken(request);
}
catch(Exception e) {
    out.write("<p>Exception! "+e+"</p>");
}

if (ssoToken != null && ssoManager.isValidToken(ssoToken)) {
%>
<p>Access allowed!</p>
<%
}
else {
%>
<p>Access denied!</p>
<%
}
%>

```

In the above code, the login and logout links use the web interface of Access Manager to verify the username and password of users. The remaining part of the code checks if the HTTP request contains a valid SSO token. If a valid SSO token is found, the user is authenticated and access is allowed.

For more details, see the following guides:

- [Sun Java System Access Manager 7.1 Documentation Collection](#)
- *Sun Java System Access Manager 7.1 Administration Guide*
- *Sun Java System Access Manager 7.1 Java API Reference*

Integrating Web Applications Into Sun OTP Web SSO Environment

There are two types of web applications that can be integrated into Web SSO. They are:

- Web applications without any authentication implementation
- Web applications with their own authentication implementation

Web Applications Without Any Authentication Implementation

Web applications that do not implement any type of authentication are protected by the Policy Agent software of Access Manager, which is installed on the host web container. For more details about configuring the Policy Agent software, see [“Configuring Policy Agent” on page 51](#). As Access Manager is integrated into Web SSO, you do not have to modify the web application.

Web Applications With Their Own Authentication Implementation

The Web SSO implementation is based on cookies and HTTP filters for redirection. To integrate a web application into Web SSO, add a filter that will redirect the user to the core Web SSO application, in case a user is not logged in. The core Web SSO application has to be extended with the authentication module. The authentication module will implement the `AuthModule` interface and provide the login and logout functionality.

Implementing the Web SSO Filter

The Web SSO filter is attached to a web application, which is part of a Web SSO. The filter intercepts all HTTP requests. When an unauthenticated HTTP request is recognized, the filter redirects the user to the Web SSO core application. When HTTP request belongs to an authenticated user, the filter does nothing. For filter API details, see package `javax.servlet`. Also see the source code, for example, `src/websso/filter/`.

For more information, see [Java Platform Enterprise Edition, v 5.0 API Specifications](#).

Extending Web SSO Core With New Authentication Module

For every web application that needs to be integrated with Web SSO, you have to implement the new authentication module. You have to modify the Web SSO core, that is, add and embed the new authentication module into the code.

The authentication module should extend the `com.sun.otp.websso.AuthModule` abstract class. The following listing illustrates it:

```

package com.sun.otp.websso.xxx;

import com.sun.otp.websso.AuthModule;
import com.sun.otp.websso.SunClusterUtils;
import com.sun.otp.websso.Util;
import com.sun.otp.websso.configuration.ConfigItem;
import java.net.HttpURLConnection;
import java.net.URL;
import javax.servlet.ServletContext;
import javax.servlet.http.Cookie;

/*****
 * Authentication module for XXX application.
 */
public class SpsAuth extends AuthModule {

    /*****/
    private static final String XXX_COOKIE = "XXXCookie";

    /*****/
    * Creates new instance of authentication module for N1 SPS.
    * @param config configuration information for this module
    * @param context servlet context
    */
    public SpsAuth(ConfigItem config, ServletContext context) {
        super(config, context);
        log("XXX: "+protocol+", "+server+", "+port);
    }

    /*****/
    * Performs login into XXX application.
    * @param username user's name
    * @param password user's password
    * @return array of cookies obtained from Lockhart that should be stored in
user's browser */
    public Cookie[] login(String username, String password) throws Exception {
        //connect to XXX application, provide credentials, and return cookies
    }

    /*****/
    * Performs logout from XXX application.
    * @return array of cookies that should be updated (deleted) in
user's browser */
    public Cookie[] logout() throws Exception {
        //connect to XXX application, do logout, and return cookies that
should be updated/deleted
    }
}

```

For filter API details, see package `javax.servlet`. Also see source code, for example, `src/webssso/filter/`. For more information, see [Java Platform Enterprise Edition, v 5.0 API Specifications](#).

Hardening Sun OTP and Hosted NEP Applications

For hardening the Sun OTP systems, Solaris Security Toolkit (SST) version 4.2 is used. SST is delivered as part of the N1 SPS plug-in. The N1 SPS plug-in contains the SST drivers for hardening and unhardening Sun OTP. As the plug-in is compiled and signed, it cannot be changed easily.

To modify a driver, do the following:

1. Create a new N1 SPS plug-in, which will inherit plans, components, and drivers from the original plug-in.
2. Insert your drivers.
3. Modify the new plug-in to use the drivers you insert.

The above steps will ensure that the plans and components from the original plug-in are reused and invoked with the new SST drivers. For more information, refer to the SST and Sun N1 SPS documentation.

- [Solaris Security Toolkit 4.2 Documentation Collection](#)
- [Sun N1 Service Provisioning System 5.2 Collection](#)

Examples of End-to-End Solution Deployment

This chapter provides examples of installing Sun OTP using GUI installation and command-line installation.

The following topics are discussed:

- [“Prerequisites for Sun OTP Installation” on page 57](#)
- [“Installing Sun OTP Using GUI Installation” on page 61](#)
- [“Installing Sun OTP Using Command-Line Installation” on page 77](#)

Prerequisites for Sun OTP Installation

The following table describes the prerequisites that you must meet before installing Sun OTP.

TABLE 9-1 Prerequisites for Sun OTP Installation

Tasks	Documentation
Determine the system requirements of Sun OTP hosts. Upgrade firmware and install patches on Sun OTP host if required. All Sun OTP hosts and storage devices must meet the minimum patch and firmware requirements.	Appendix B, “Sun OTP Hardware and Software Requirements,” in <i>Sun Open Telecommunications Platform 2.0 Installation Guide</i>
Set up a naming service such as DNS, NIS, NIS+, or /etc/hosts and all host names and IP addresses must be set up on that naming service.	<i>System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)</i>
Install Solaris 10 Update 3 OS on the Sun OTP provisioning server and on each Sun OTP host.	Installing Solaris OS from the Bootable DVD
Prepare the Sun OTP hosts for Sun OTP installation.	Preparing the Sun OTP Hosts

TABLE 9-1 Prerequisites for Sun OTP Installation (Continued)	
Tasks	Documentation
Set up the self-contained Sun OTP provisioning server.	Setting up the Self-Contained Sun OTP Provisioning Server
Prepare the Sun OTP host plan worksheet. Plan information includes items such as logical host names, IP addresses for each Sun OTP host, the clustered Sun OTP system name, IPMP options, zone information, and more. Using the worksheets will assist you during installation and configuration, and reduce the chance for errors.	Appendix A, “Sun OTP Plan Worksheet”

Prerequisite Tasks for Sun OTP Installation

The following prerequisite tasks must be met before you install Sun OTP.

- [Installing Solaris OS from the Bootable DVD](#)
- [Preparing the Sun OTP Hosts](#)
- [Setting up the Self-Contained Sun OTP Provisioning Server](#)

▼ To Install Solaris OS from the Bootable DVD

- **Install Solaris OS from the bootable OTP DVD.**

Refer to *Solaris 10 11/06 Installation Guide: Basic Installations* or *Solaris 10 11/06 Installation Guide: Solaris Flash Archives (Creation and Installation)* for more information.

- **On SPARC systems**

- a. **Insert the bootable Sun OTP DVD.**

Note – The bootable DVD for SPARC contains the hybrid flash image that can be used for deploying the sun4v and sun4u hardware architecture platforms.

- b. **Start the installation from the bootable DVD by typing the following command:**

`boot cdrom`

This will prompt you for an interactive Solaris installation.

- c. **Respond to the system identification and configuration screens.**

- d. **From the Solaris Interactive Installation screen, select F4_Flash install.**

- e. From the Flash Archive Retrieval Method screen, place the X symbol on Local File and press F2_Continue.
- f. From the Flash Archive Addition screen, provide the `/cdrom/Solaris_10/otp_sparc.flarc` path and then press F2_Continue.
Continue to respond to the remaining screens until the installation starts and completes.
- g. When the system reboots, insert the installation DVD.

- On x64 systems



Caution – Installation does not ask for the partitioning or image name. The first disk is chosen automatically and OTP default disk formatting is applied. The existing data, if any, is overwritten.

- a. Select CDROM as a boot device in BIOS.
- b. Respond to the system identification and configuration screens.
- c. When the system reboots, insert the installation DVD.

▼ To Prepare the Sun OTP Hosts

- 1 Log in as root (`su - root`) to the Sun OTP host.

- 2 Install the remote agent on the Sun OTP host.

```
/opt/SUNWotp/cli/setupRemoteAgent -c RAW|SSH mediadir
```

mediadir is the fully qualified path name to the Open Telecommunications Platform installation source directory. The media directory is the path where the installation DVD is mounted.

Repeat this step on each Sun OTP host. All the remote agents on the cluster must use the same connection type (SSH or RAW).

- 3 (Applicable for SSH connection type) Configure SSH for the remote agent.

- a. Set a password for the `n1spsotp` user in the global zone on the Sun OTP host.

```
/usr/bin/passwd n1spsotp
```

- 4 (Applicable for SSH connection type) Initialize the SSH keys.

- a. Log in as `spsotp` user (`su - spsotp`) to the Sun OTP provisioning server.

- b. **Append the ssh public key of `spsopt` user from the Sun OTP provisioning server to authorized ssh keys located in the home directory of `n1spsopt` user on each Sun OTP host.**

```
cat /var/otp/.ssh/id_rsa.pub | ssh n1spsopt@hostname "tee >>
/export/home/n1spsopt/.ssh/authorized_keys2"
```

hostname is the host name of Sun OTP host.

▼ To Set up the Self-Contained Sun OTP Provisioning Server

Self-contained Sun OTP provisioning server is installed on the first host of the cluster.

Self-contained Sun OTP provisioning server uses a specific logical host name and IP address defined at the beginning of the Sun OTP installation. However, to make Sun OTP application provisioning service highly available, the server is moved to a different logical host name (logical host name for management and provisioning services) at the end of “Configure Components as Highly Available Services” plan. After “Configure Components as Highly Available Services” plan is finished, this new logical host name has to be used to access the Sun OTP provisioning server.

- 1 **Log in as root (`su - root`) to the first host of the cluster.**

- 2 **Add the logical host name and the logical IP address to the `/etc/hosts` file.**

This logical host name and the logical IP address is used in the next step to set up the Sun OTP provisioning server.

- 3 **Set up the Sun OTP provisioning server in the first host of the cluster.**

```
/opt/SUNWotp/cli/setupExternalInstallServer -M mediadir -N nodetype -R RAW|SSH -P
passwordfile -L logicalhost -I logicalip
```

mediadir is the fully qualified path name to the Open Telecommunications Platform installation source directory. The media directory is the path where the installation DVD is mounted.

nodetype can either be `single` (for stand-alone deployments) or `first` (for clustered deployments).

`-R` is the connection type between Sun OTP application provisioning service master server and the remote agent. `RAW` and `SSH` are the possible connection types between Sun OTP application provisioning service master server and the remote agent.

logicalhost and *logicalip* are unused logical host names and logical IP addresses.

passwordfile is the absolute path of the password file. You can create this file in your home directory. The password file must contain a line with a valid password for the Sun OTP provisioning server. The password can be 8 to 12 alphanumeric characters.

Note – Once you set up the Sun OTP provisioning server with this password, the user name to access the Sun OTP provisioning server is `otpadmin` and the password is the password in the password file.

The password specified in this step and the password that will be specified during the Setup Configuration plan must be the same.

Installing Sun OTP Using GUI Installation

This section describes Sun OTP installation on a clustered system using the GUI installation method.

Note – Sun OTP can also be installed on a stand-alone system with logical domains. In this configuration, all the Sun OTP components are deployed inside a logical domain. This logical domain must be created before installing Sun OTP.

The following topics are discussed:

- [“To Install Sun OTP on a Clustered System” on page 61](#)
- [“To Configure the Quorum Disk on a Two-Host Cluster” on page 76](#)

▼ To Install Sun OTP on a Clustered System

- It is recommended that the shared storage is configured to provide redundancy of shared disks using hardware RAID (RAID1 or RAID5).
- If the above feature is not available in the storage, it is recommended to use software RAID (RAID1 or RAID5) for shared storage provided by Solaris Volume Manager. See *Solaris Volume Manager Administration Guide* for more information. In this case, the shared storage needs to be set up manually.
- In this procedure, the cluster host set refers to all the Sun OTP hosts in the cluster and the additional cluster host set refers to all the additional Sun OTP hosts in the cluster except the first Sun OTP host.

Before You Begin Make sure that you complete the tasks described in [“Prerequisite Tasks for Sun OTP Installation” on page 58](#).

- 1 **Open a browser and log in to the Sun OTP application provisioning service on the Sun OTP provisioning server.**

Go to the URL `https://install server:9090` where *install server* is the IP address or the fully-qualified name of the Sun OTP provisioning server.

2 Type the user name and password.

The user name is otpadmin. The password is the password provided in the password file while setting up the Sun OTP provisioning server.

3 Add Sun OTP hosts to the Sun OTP provisioning server. Repeat this step on each Sun OTP host.

a. Click Host Setup in the left menu to display the Host Setup page.

b. Click hosts in the central menu to display the hosts page.

c. In the host field, type the name of the new Sun OTP host.

d. (Optional) In the description field, type a description of the new Sun OTP host.

e. Click create.

The host details edit page is displayed.

f. To include the remote agent, select the include remote agent on this physical host check box.

g. Choose SSH (encrypted) or RAW from the connection type drop-down list.

SSH (encrypted) is the recommended value.

h. Type the host name in the ip address or name field.

i. Type the appropriate entry according to the connection type.

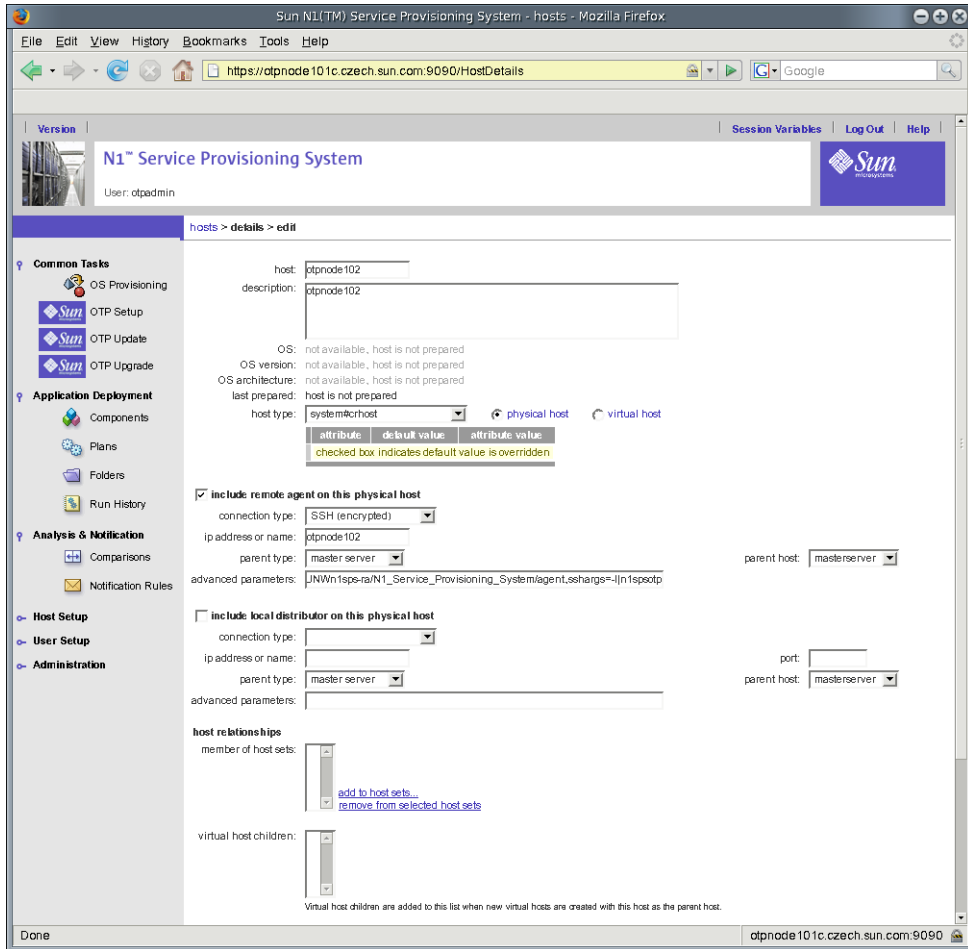


FIGURE 9-1 Add OTP Hosts to the Provisioning Server Screen

- If you chose SSH (encrypted) from the connection type drop-down list, type the following entry in the advanced parameters field.

`cprefix=/opt/SUNWn1sps-ra/N1_Service_Provisioning_System/agent,sshargs=-l|n1sps`

- If you chose RAW from the connection type drop-down list, type 7000 in the port field.

j. Click add to host sets.

k. Select `com.sun.solaris#SolarisHostSet` from the add host sets drop-down list.

l. Click add host to selected host sets.

m. Scroll to the bottom of the page and click save.

The host preparation is in progress and Sun OTP host is added to the hosts list on the Sun OTP provisioning server.

n. In the hosts page, click the name of the host that is added.

o. Click update remote agent.

p. After successful update, click prepare remote agent.

The host is prepared for provisioning.

4 (Optional but recommended) Create host sets.

It is recommended to create two host sets. One host set will contain all the Sun OTP additional hosts in the cluster and the other host set will contain the previously created host set and the first cluster host.

a. Click Host Setup.

b. Click host sets to display the host sets page.

c. Click create.

d. Type the name of the host set in the host set field.

e. Click select from list.

f. Select the hosts that you want to add to the host set.

Note – Select only the Sun OTP host names and not the zone host names.

g. Click add hosts to main window.

h. Click add host.

i. Click save.

5 Click OTP Setup to display the Sun Open Telecommunications Platform installation steps page.

6 Configure Sun OTP installation plan variables. Repeat this step on each Sun OTP host.

a. Click Setup Configuration and click run.

The SetupConfig screen appears.

- b. Type the host name on which you want to install Sun OTP in the target host field.
- c. Enter the password for the `otpadmin` user in the Enter otpadmin password field.
- d. Confirm the password in the Confirm otpadmin password field.
- e. Click select from list under variable settings.
The select variable setting from list... screen appears.
- f. At the bottom of the select variable setting from list... screen, click create set to create a new variable set.
- g. Type a new variable set name in the Set Name field.
- h. Click the check boxes for the appropriate plan variables for which you want to enter the values.



FIGURE 9-2 Configure OTP Plan Variables Screen

- i. Type the values for the appropriate plan variables in the text fields. Refer to [Appendix A, “Sun OTP Plan Worksheet”](#) to determine the values for these variables.
- j. Click save to save the variable set.
- k. Close the select variable setting from list... screen.
- l. In the SetupConfig screen, click the drop-down list under variable settings, and choose the new variable set.
- m. Click run plan (includes preflight).

7 Install the OS patches on the additional Sun OTP hosts.

a. Click Patch and click run.

The Patch screen appears.

b. Choose the additional cluster host set from the target host set drop-down list.

c. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

8 Install the OS patches on the first Sun OTP host.

a. Click Patch and click run.

The Patch screen appears.

b. Type the first Sun OTP host name in the target host field.

c. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

When the plan completes, wait for the first Sun OTP host to boot into multi-user mode.

9 Install and configure the Sun OTP high availability service on the first Sun OTP host.

a. Click Install Availability Service and click run.

The InstallAvailability1 screen appears.

b. Type the first Sun OTP host name in the target host field.

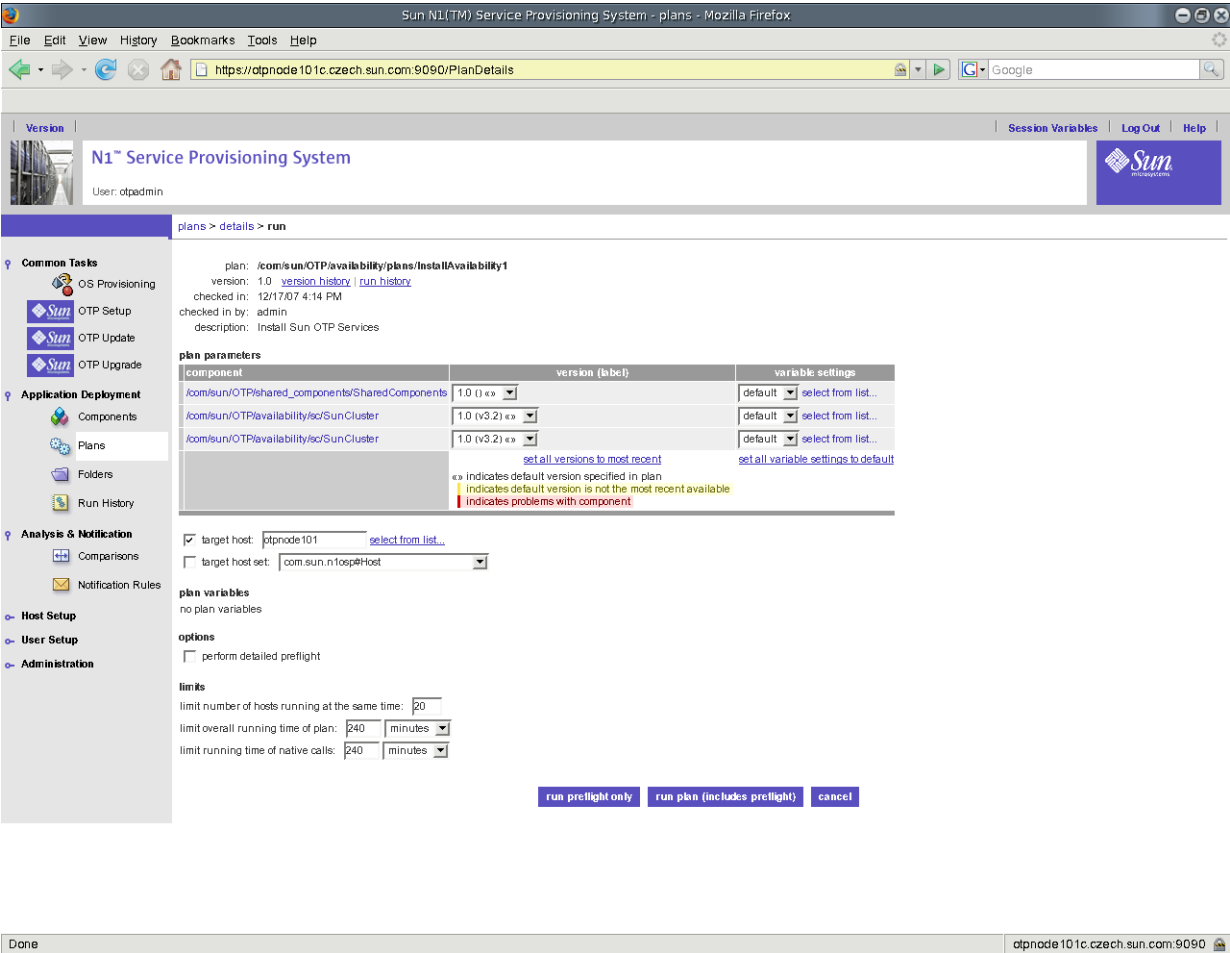


FIGURE 9–3 Install Availability Service Screen

- c. **Click run plan (includes preflight).**
The page is reloaded and a progress bar is displayed during the process.
When the plan completes, wait for the first Sun OTP host to reboot.
- d. **Click Configure Availability Service and click run.**
The InstallAvailability2 screen appears.
- e. **Type the first Sun OTP host name in the target host field.**

f. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

10 Install and configure the Sun OTP high availability service on the additional Sun OTP hosts.

a. Click Install Availability Service and click run.

The InstallAvailability1 screen appears.

b. Choose the additional cluster host set from the target host set drop-down list.

c. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

When the plan completes, wait for the Sun OTP host to reboot.

d. Click Configure Availability Service and click run.

The InstallAvailability2 screen appears.

e. Choose the additional cluster host set from the target host set drop-down list.

f. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

11 (Applicable for zones) Create and configure a non-global zone on all the Sun OTP hosts.

a. Click Configure Solaris Containers and click run.

The InstallZones screen appears.

This step also installs the remote agent in the non-global zone.

b. Choose the cluster host set from the target host set drop-down list.

c. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

12 (Applicable for zones) Configure SSH for the remote agent in the newly created non-global zone on the Sun OTP host.

Note – Perform this step only when you have chosen SSH (encrypted) connection type between Sun OTP application provisioning service master server and the remote agent.

a. Log in to the non-global zone on the Sun OTP host.

`zlogin zonename`

Where *zonename* is the name of the non-global zone.

- b. **Set a password for n1spsotp user.**
- c. **Log in as spsotp user (su - spsotp) on the Sun OTP provisioning server.**
- d. **Append the ssh public key of the spsotp user from the Sun OTP provisioning server to the authorized ssh keys located in the home directory of the n1spsotp user on each non-global zone on Sun OTP host.**

```
cat /var/otp/.ssh/id_rsa.pub | ssh n1spsotp@zonehostname "tee >>
/export/home/n1spsotp/.ssh/authorized_keys2"
```

where *zonehostname* is the zone host name of Sun OTP host.

13 Create shared storage on the clustered Sun OTP system.

The shared storage is used for high availability of Sun OTP system management service and Sun OTP application provisioning service. These services are installed on the shared storage. If one host fails, the shared storage is mounted onto the other host and services will be restarted.

The shared storage will contain the otp-system-rg resource group.

- a. **Click Setup Shared Storage and click run.**

The CreateSharedStorage screen appears.

Ensure that the shared storage has at least 4 GB space.

- b. **Click select from list... under variable settings.**

The select variable setting from list... screen appears.

- c. **At the bottom of the select variable setting from list... screen, click create set to create a new variable set.**

- d. **Type the variable set name in the Set Name field.**

- e. **Provide values for the following parameters:**

- **rootDisk** - Name of the root disk.
- **diskSlice** - Name of the disk slice where Solaris Volume Manager (SVM) database is created.
- **mntPoint** - Name of the mount point. The value must be `/var/jfs`.
- **diskGroup** - Name of the disk group. For example, `sm-dg`.
- **hostList** - List of all the Sun OTP hosts. Separate the host names by a colon. For example, `hostname1:hostname2`.
- **metasetDID** - Name of the valid shared did. For example, `d5`.

Type the command `cldevice list -v` to determine the shared did.

- f. Close the select variable setting from list... screen.**
 - g. In the CreateSharedStorage screen, click the drop-down list under variable settings, and choose the new variable set.**
 - h. Choose the cluster host set from the target host set drop-down list.**
 - i. Click run plan (includes preflight).**
- 14 Install and configure the Sun OTP system management service in the global zone on all the Sun OTP hosts.**
 - a. Click Install System Management Service and click run.**

The InstallManagement screen appears.

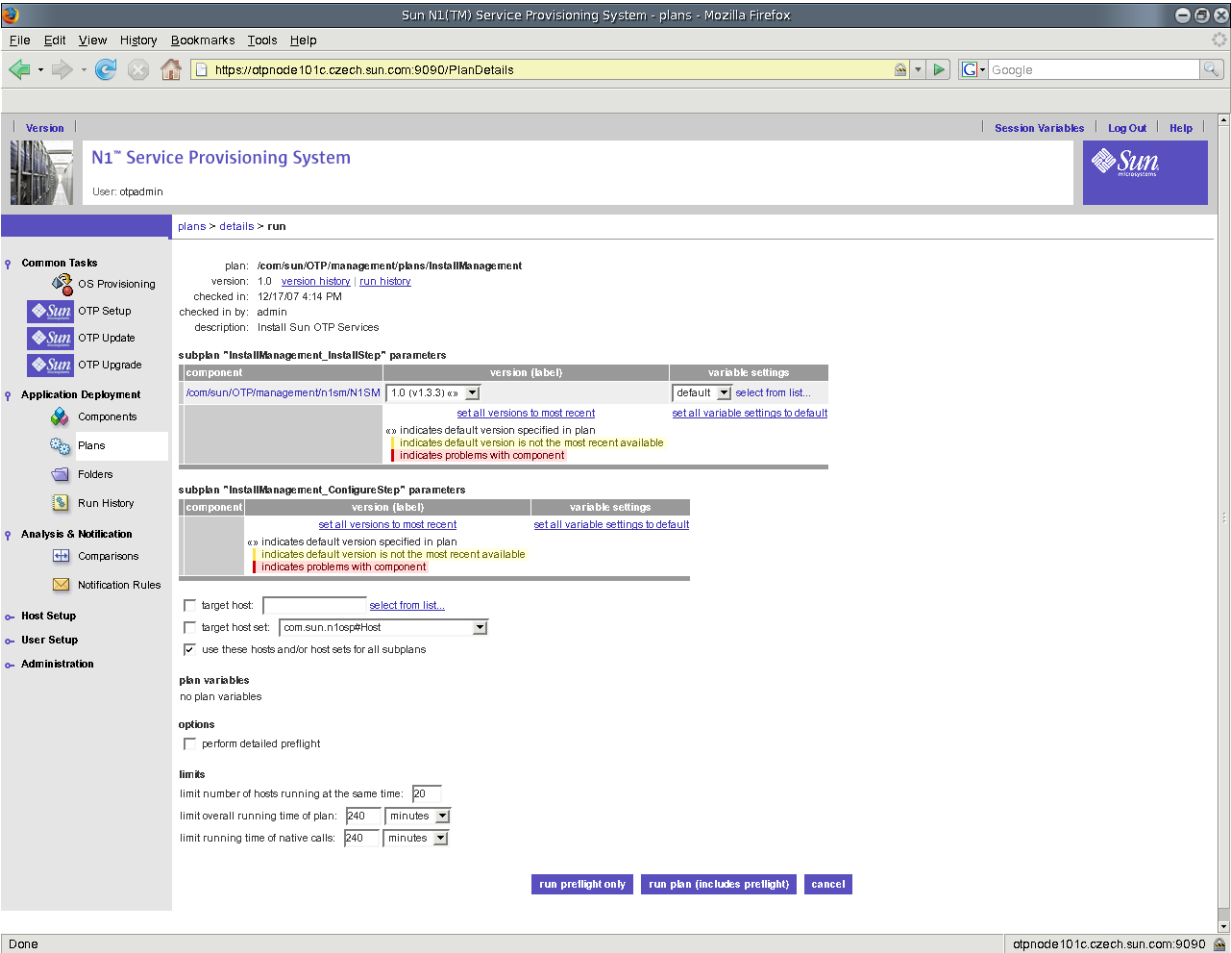


FIGURE 9-4 Install System Management Service Screen

b. Choose the cluster host set from the target host set drop-down list.

c. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

15 Install and configure the Sun OTP application provisioning service in the global zone on all the Sun OTP hosts.

a. Click Install Provisioning Service and click run.

The InstallProvisioning screen appears.

b. Choose the cluster host set from the target host set drop-down list.

c. Click run plan (includes preflight).

The page is reloaded and a progress bar is displayed during the process.

16 (Applicable for zones) Update and prepare the new zone host with the Sun OTP provisioning server.

a. Click Host Setup in the left panel.

b. Click hosts.

c. In the hosts page, click the name of the host that is added.

d. Click update remote agent.

e. After successful update, click prepare remote agent.

17 (Applicable for zones) Configure Sun OTP installation plan variables for the new zone host.

a. Click OTP Setup to display the Sun Open Telecommunications Platform installation steps page.

b. Click Setup Configuration and click run.

c. Choose the same variable set that you chose in [Step 6](#).

d. Type the zone host name in the target host field.

e. Click run plan (includes preflight).

18 Install and configure the Sun OTP security service on all the Sun OTP hosts.

a. Click Install Security Service and click run.

The InstallSecurity screen appears.

b. Choose the cluster host set from the target host set drop-down list.

The page is reloaded and a progress bar is displayed during the process.

a. Click **Configure Components** and click **run**.

Sun Open Telecommunications Platform 2.0 Developer's Guide • February 2008

This step creates and starts resource groups for Sun OTP system management service, Sun OTP application provisioning service, and Sun OTP security service. This step also configures and starts master-to-master replication (MMR).

- b. **Choose the cluster host set from the target host set drop-down list.**
- c. **Click run plan (includes preflight).**

The page is reloaded and a progress bar is displayed during the process.

Note – Self-contained Sun OTP provisioning server uses a specific logical host name and IP address defined at the beginning of the Sun OTP installation. However, to make Sun OTP application provisioning service highly available, the server is moved to a different logical host name (logical host name for management and provisioning services) at the end of this plan. After this plan is finished, this new logical host name has to be used to access the Sun OTP provisioning server.

20 Install Web SSO.

Note – When you install Sun OTP through GUI installation from a self-contained Sun OTP provisioning server, you need to verify the completion status of the previous plan (Configure and Enable HA plan) through the `/var/OTP/SUNWotp.log` file before running the Install Web SSO plan. The file must have the line that reads “enableHA - COMPLETED”.

- a. **Click Install Web SSO and click run.**
The InstallWebSSO screen appears.
- b. **Choose the cluster host set from the target host set drop-down list.**
- c. **Click run plan (includes preflight).**

Note – Monitor the `/var/OTP/SUNWotp-debug.log` file to check whether the resource group `otp-system-rg` has been restarted. If the resource group has not been restarted, restart the resource group manually by typing the following command on any host of the cluster.

```
/usr/cluster/bin/clrg online otp-system-rg
```

The installation log files, input files generated for the plans, installation registry information, and the debug log files are stored in the `/var/OTP` directory.

See Also On a two-host cluster, if you chose no for `quorumAutoConfiguration` during variable set creation, you must manually select and configure the quorum disk as described in [“To Configure the Quorum Disk on a Two-Host Cluster”](#) on page 76.

▼ To Configure the Quorum Disk on a Two-Host Cluster

On a two-host cluster, if you chose no for Quorum Auto Configuration in the `input_otp.dat` file, you must manually select and configure the quorum disk as described in this procedure.

Note – The following substeps apply only to a two-host cluster. If you are setting up a three-host or more clustered Sun OTP system, this procedure is optional.

- 1 **Open a separate terminal window and log in as root to the first Sun OTP host.**
- 2 **Type the following command to display the cluster disk information.**

```
/usr/cluster/bin/cldev list -v
```

DID	Device	Full Device Path
d1		pci44-otp1:/dev/rdisk/c0t8d0
d1		pci44-otp2:/dev/rdisk/c1t8d0
d2		pci44-otp1:/dev/rdisk/c0t9d0
d2		pci44-otp2:/dev/rdisk/c1t9d0
d3		pci44-otp1:/dev/rdisk/c1t10d0
d4		pci44-otp1:/dev/rdisk/c1t11d0
d4		pci44-otp2:/dev/rdisk/c0t11d0
d5		pci44-otp1:/dev/rdisk/c2t1d0
d6		pci44-otp1:/dev/rdisk/c2t3d0
d7		pci44-otp1:/dev/rdisk/c3t0d0
d9		pci44-otp2:/dev/rdisk/c2t3d0
d10		pci44-otp2:/dev/rdisk/c3t0d0

In the above example, disks d1 and d2 are shared by both hosts of the two-host cluster. The quorum disk must be a shared disk.

- 3 **Configure a quorum disk.**

```
/usr/cluster/bin/clq add shared disk ID
```

Where *shared disk ID* is a shared disk ID.

For example, to configure d4 as a quorum disk, type the following command.

```
/usr/cluster/bin/clq add d4
```

- 4 **Reset the two-host cluster to normal mode.**

```
/usr/cluster/bin/clq reset
```

Installing Sun OTP Using Command-Line Installation

This section describes Sun OTP installation on a clustered system using the command-line installation method.

Note – The procedures for command-line installation are examples and are provided only for demonstration purposes.

▼ To Install Sun OTP on a Clustered System

- It is recommended that the shared storage is configured to provide redundancy of shared disks using hardware RAID (RAID1 or RAID5).
- If the above feature is not available in the storage, it is recommended to use software RAID (RAID1 or RAID5) for shared storage provided by Solaris Volume Manager. See *Solaris Volume Manager Administration Guide* for more information. In this case, the shared storage needs to be set up manually.

Before You Begin Make sure that you complete the tasks described in [“Prerequisite Tasks for Sun OTP Installation” on page 58](#).

1 Log in as root (su - root) to the self-contained Sun OTP provisioning server.

2 Copy the input_otp.dat file to a local non-temporary directory.

```
cp /opt/SUNWotp/cli/templates/input_otp.dat /export/
```

3 Edit the /export/input_otp.dat file.

Type the values for the appropriate plan variables in the text fields. Refer to [Appendix A, “Sun OTP Plan Worksheet”](#) to determine the values for these variables.

4 Set up the Sun OTP configuration on all the Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i S -f /export/input_otp.dat -o "-P passwordfile"
```

This command specifies the Sun OTP deployment parameters and validates these parameters provided in the input_otp.dat file.

5 Install the OS patches on all the Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i P -f /export/input_otp.dat
```

When the command completes, wait for the Sun OTP hosts to boot into multi-user mode.

6 Install and configure the Sun OTP high availability service in the global zone on the first Sun OTP host.

```
/opt/SUNWotp/cli/deploy_otp -i a -f /export/input_otp.dat -o "-N first"
```

When the command completes, wait for the Sun OTP host to reboot completely and then type the following command:

```
/opt/SUNWotp/cli/deploy_otp -c a -f /export/input_otp.dat -o "-N first"
```

7 Install and configure the Sun OTP high availability service in the global zone on the additional Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i a -f /export/input_otp.dat -o "-N additional"
```

When the command completes, wait for the Sun OTP host to reboot completely and then type the following command:

```
/opt/SUNWotp/cli/deploy_otp -c a -f /export/input_otp.dat -o "-N additional"
```

8 (Applicable for zones) Create and configure a non-global zone on all the Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i z -f /export/input_otp.dat
```

This command takes the zone-related parameters in the `input_otp.dat` file and creates a non-global zone. This command also installs the remote agent in the non-global zone on all the Sun OTP hosts.

9 (Applicable for zones) Configure SSH for the remote agent in the newly created non-global zone on all the Sun OTP hosts.

a. Log in to the non-global zone on the Sun OTP host.

```
zlogin zonename
```

Where *zonename* is the name of the non-global zone.

b. Set a password for `n1spsotp` user in the global zone on the Sun OTP host.

c. Log in as the `spsotp` user (`su - spsotp`) on Sun OTP provisioning server.

d. Append the ssh public key of the `spsotp` user from the provisioning server to the authorized ssh keys located in the home directory of the `n1spsotp` user on each Sun OTP host.

```
cat /var/otp/.ssh/id_rsa.pub | ssh n1spsotp@zonehostname "tee >>  
/export/home/n1spsotp/.ssh/authorized_keys2"
```

where *zonehostname* is the zone host name of Sun OTP host.

10 Create shared storage on the clustered Sun OTP system.

The shared storage is used for high availability of Sun OTP system management service and Sun OTP application provisioning service. These services are installed on the shared storage. If one host fails, the shared storage is mounted onto the other host and services will be restarted.

The shared storage will contain the `otp-system-rg` resource group.

```
/opt/SUNWotp/cli/deploy_otp -i d -f /export/input_otp.dat -o "-D did -L hostlist -G diskgroup -M mountpoint"
```

Where *did* is the valid shared did. For example, `d5`.

hostlist is the list of all Sun OTP hosts connected to shared storage. Separate the host names by a colon. For example, `hostname1:hostname2`.

diskgroup is the disk group. For example, `sm-dg`.

mountpoint is the mount point.

To determine the shared did, type the following command:

```
/usr/cluster/bin/cldev list -v
```

Choose the did that is shared among the Sun OTP hosts.

11 Install and configure the Sun OTP system management service in the global zone on all the Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i m -f /export/input_otp.dat
```

12 Install and configure the Sun OTP application provisioning service in the global zone on all the Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i p -f /export/input_otp.dat
```

13 Install and configure the Sun OTP security service on all the Sun OTP hosts.

```
/opt/SUNWotp/cli/deploy_otp -i s -f /export/input_otp.dat
```

14 Configure and enable high availability for Sun OTP services on the first Sun OTP host.

```
/opt/SUNWotp/cli/deploy_otp -c h -f /export/input_otp.dat
```

This command creates and starts resource groups for Sun OTP system management service, Sun OTP application provisioning service, and Sun OTP security service. This command also configures and starts master-to-master replication (MMR).

Note – Self-contained Sun OTP provisioning server uses a specific logical host name and IP address defined at the beginning of the Sun OTP installation. However, to make Sun OTP application provisioning service highly available, the logical hostname that was previously used will be released upon successful completion of the Configure and Enable HA service plan, and the Sun OTP application provisioning service will be accessible through the Management and Provisioning logical hostname and IP address.

15 Install Web SSO.

```
/opt/SUNWotp/cli/deploy_otp -i o -f /export/input_otp.dat
```

Note – Monitor the `/var/OTP/SUNWotp-debug.log` file to check whether the resource group `otp-system-rg` has been restarted. If the resource group has not been restarted, restart the resource group manually by typing the following command on any host of the cluster.

```
/usr/cluster/bin/clrg online otp-system-rg
```

The installation log files, input files generated for the plans, installation registry information, and the debug log files are stored in the `/var/OTP` directory.

See Also On a two-host cluster, if you chose no for `quorumAutoConfiguration` during variable set creation, you must manually select and configure the quorum disk as described in [“To Configure the Quorum Disk on a Two-Host Cluster”](#) on page 76.

Sun OTP Plan Worksheet

This appendix provides descriptions of Sun OTP installation variables and plan worksheet.

The following topics are discussed:

- [“Sun OTP Plan Settings Description” on page 81](#)
- [“Sun OTP Plan Worksheet” on page 85](#)

Sun OTP Plan Settings Description

The following list describes each of the Sun OTP plan settings that are used by the Sun OTP installation. These settings need to be specified in the Setup Configuration plan during the installation.

- **mediaDirectory**
The fully-qualified path name to the Sun Open Telecommunications Platform installation source directory.
- **clusterName**
Name of the cluster. The length of the cluster name has to be less than 19 characters.
- **mgmtHost**
Logical host name used by the Sun OTP system management service. The logical host name must correspond to the management logical IP address.
- **mgmtIP**
Logical IP address used by the Sun OTP system management service. The IP address must be an unused IP address.
- **jesHAHost**
Host name for Sun OTP security service shared address.
- **jesHANodeList**
List of host names and respective zones on which the Sun OTP security service is running.

For deployment without zones, use the following syntax:

h1_hostName+h2_hostName...

For deployment with zones, use the following syntax:

h1_hostName:h1_zoneName+h2_hostName:h2_zoneName...

- **mmrHostList**

List of host names or zone host names between which the master-to-master replication of data between directory server instances are provided.

For deployment without zones, use the following syntax:

h1_hostName+h2_hostName...

For deployment with zones, use the following syntax:

h1_zoneHostname+h2_zoneHostname...

- **applyAllPatches**

Specifies whether all patches or only mandatory patches are to be installed. The default value is yes, which specifies that all patches must be installed. To install only mandatory patches, specify no.

- **spsRAConnectionType**

Specifies the connection type between Sun OTP application provisioning service master server and the remote agent. The values can be SSH or RAW. The default and recommended value is SSH.

This variable must match the connection type that is provided while setting up the Sun OTP provisioning server and installing the remote agent.

- **hostName**

The host name of the Sun OTP host. The length of the host name has to be less than 19 characters.

- **hostType**

Type of host in the cluster. The possible values are `single`, `first`, and `additional`. For stand-alone host, the host type is `single`. For clustered configuration, the host type is `first` for the first host in the cluster and `additional` for the remaining hosts in the cluster.

- **sponsorNode**

The name of the first Sun OTP host in a clustered Sun OTP system. The first Sun OTP host is the sponsoring node for all the additional Sun OTP hosts. This setting is required when installing the Sun Open Telecommunications Platform on two or more hosts.

This variable is required only for hosts of host type `additional`.

- **autoConfigureIPMP**

Setting to determine whether IPMP is configured automatically. The Possible values are yes and no. The default value is no.

You should configure all physical interfaces of a multipathing group with a test IP address. Test addresses are required to detect failures.

If you set `autoConfigureIPMP=yes`, then you must also specify the following values:

- **secondaryInterface**

Interface used as the failover interface if a fault is detected on the primary interface.

- **secondaryIP**

IP address of the secondary interface that is used for failover. This variable is the same as **testIPAddress** but used for the secondary interface.

- **testIPAddress**

An unused IP address that is to be assigned as a routable, no-failover, and deprecated test IP address to the adapter. IP network multipathing uses test addresses to detect network path failures, switch port faults, and partial network equipment outages.

- **privateInterface1**

The first private network interface on Sun OTP host. Required when **hostType** is `first` or `additional`.

- **privateInterface2**

The second private network interface on Sun OTP host. Required when **_hostType** is `first` or `additional`.

- **transportTypeInterface1**

The transport type of the first private interconnect adapter on Sun OTP host. The required value is `dlpi`. Do not change the value.

For more information, refer to *dlpi(7P)* man page.

- **transportTypeInterface2**

The transport type of the second private interconnect adapter on Sun OTP host. The required value is `dlpi`. Do not change the value.

For more information, refer to *dlpi(7P)* man page.

- **nodeAuthentication**

Setting to establish the authentication policies for hosts. The possible values are `sys` and `des`. The default value is `sys`.

- **quorumAutoConfiguration**

Note – Quorum automatic configuration applies only to two-host clusters.

Quorum automatic configuration provides an option to enable or disable automatic configuration of the quorum device in a two-host only clustered Sun OTP system.

The possible values are `yes` and `no`. The default value is `yes`.

Note – If this value is set to no, a manual administrative procedure is required to configure the quorum disk in a two-host clustered Sun OTP system. The cluster must be manually reset from the install mode to the normal mode. For details about how to configure quorum disks, refer to the `scconf` command documentation in `scconf(1M)` man page.

- **rootDisk**
Name of the root disk that is used to store the Solaris Volume Manager (SVM) database.
- **diskSlice**
The disk slice where the Solaris Volume Manager (SVM) state database replicas are stored. The default value is `s7`. Refer to *Solaris Volume Manager Administration Guide* for more information.
- **zoneName**
Name of the non-global zone for the Sun OTP security service. This variable has to be filled or remain empty for deployment with or without zones respectively.
- **zoneInterface**
Network interface to be used for the non-global zone, typically the primary interface of the host.
- **zoneIPAddress**
IP address of the non-global zone.
- **zoneHostName**
Host name of the non-global zone corresponding to **zoneIPAddress**.
- **zonePath**
Path to store the non-global zone.
- **zoneMask**
Network mask of the non-global zone.
- **zoneDefaultRoute**
Network default route of the non-global zone. **zoneDefaultRoute** must be on the same logical subnet as **zoneIPAddress**.
- **managementInterface**
Name of the network interface used for Sun OTP system management services. The name of the interface depends on the platform type.
- **provisioningInterface**
Name of the network interface used for Sun OTP application provisioning services. The name of the interface depends on the platform type.
- **domainName**

Domain name used by the Sun OTP security service.

- **ssoCookieDomain**

Domain name for the Web SSO cookies. The Domain name must start with a dot (.) symbol.

Sun OTP Plan Worksheet

The following table lists the settings that you need to provide for each host during installation and configuration of the Sun Open Telecommunications Platform on a clustered Sun OTP system.

Tip – Print a copy of the following table for each host and then fill out the required information to use when installing and configuring the Sun Open Telecommunications Platform on a clustered Sun OTP system.

TABLE A-1 Sun OTP Plan Worksheet

Setting Name	Example
mediaDirectory	/cdrom/otp_20_dvd/otp2.0
clusterName	otp-cluster
mgmtHost	otp-node1b
mgmtIP	10.18.144.85
jesHAHost	otp-node1c
jesHANodeList	otp-node1: jeszone+otp-node2: jeszone
mmrHostList	otp-node1d+otp-node2d
applyAllPatches	yes
spsRAConnectionType	SSH
hostName	otp-node1
hostType	first
sponsorNode	h2_sponsorNode=OTPfirshost
autoConfigureIPMP	yes
secondaryInterface	bge1
secondaryIP	10.18.144.82
testIPAddress	10.18.144.81

TABLE A-1 Sun OTP Plan Worksheet (Continued)

Setting Name	Example
privateInterface1	bge2
privateInterface2	bge3
transportTypeInterface1	dlpi
transportTypeInterface2	dlpi
nodeAuthentication	sys
quorumAutoConfiguration	yes
rootDisk	c0t0d0
diskSlice	s7
zoneName	jeszone
zoneInterface	bge0
zoneIPAddress	10.18.144.86
zonePath	/zone/jeszone
zoneHostname	otp-node1d
zoneMask	255.255.255.0
zoneDefaultRoute	10.18.144.1
managementInterface	bge0
provisioningInterface	bge0
domainName	czech.sun.com
ssoCookieDomain	.sun.com

Sun OTP Log Files

This appendix lists the useful log files and their locations.

Log Files

OTP Installation Log Files

There are two OTP installation log files.

- /var/OTP/SUNWotp.log
- /var/OTP/SUNWotp-debug.log

Log File Locations

The following table provides a list of log file locations.

TABLE B-1 Log File Locations

Log File	Description
/var/OTP	Installation log files, input files generated for the plans, installation registry information, and the debug log files
/var/SUNWotp	Security services log files and run-time information in the zones
/etc/opt/SUNWotp	Sun OTP Web SSO configuration files
/var/otp, /var/js	Fail-over services installed on shared storage
/opt/SUNWotp	Sun OTP packages, security services base

TABLE B-1 Log File Locations (Continued)

Log File	Description
/var/opt/SUNWotp	Security services runtime-instances, logs

Index

A

- Adapting your application for management
 - agent builder, 38
 - C/C++, 38
 - scripts, 38
- Adapting your application for provisioning
 - calling the installation script, 33-34
 - moving the application deployment logic, 34-36
 - using N1 SPS provisioning features, 36-37

B

- Bootable DVD, Solaris OS installation, 58-59

C

- Calling N1 SPS
 - C/C++, 41-43
 - Java, 43-45
 - shell scripts, 39-41
- CLI installation, Sun OTP, 77-80
- Component versions, Sun OTP, 10-11
- Components, Sun OTP, 23-25
- Configuration, IPMP, 82
- Configuring, policy agent, 51-53
- Control points, Sun OTP, 48
- CORBA, 17
- Customization, Sun OTP, 30-31

D

- Deployment methods, Sun OTP, 49-50
- DNS, 19

E

- end-to-end solution deployment, Sun OTP, 47-50

F

- FC/FCP, 19

G

- GUI installation, Sun OTP, 61-76
- Guides, Sun OTP, 11

H

- Hardening, Sun OTP and NEP applications, 56
- Host preparation, Sun OTP, 59-60

I

- IIOP, 19
- Installation log files, Sun OTP, 87-88
- Integrating web applications, Sun OTP Web SSO, 54-56

Interfaces, Sun OTP, 17-18
IPMP, configuration, 82
iSCSI, 19

J

Java interfaces, 17
Java RMI, 19

N

NEP applications, hardening, 56

P

PAM, 17
Plan settings, Sun OTP, 81-85
Plans, Sun OTP, 25-30
Plug-in, Sun OTP, 30
Policy agent, configuration, 51-53
POSIX, 17
Prerequisite tasks, Sun OTP installation, 58-61
Prerequisites, Sun OTP installation, 57-61
Product mapping, Sun OTP, 10-11
Protocols, Sun OTP, 18-19

R

RMI, 19
RMI-IIOP, 19

S

Securing web applications, Sun OTP, 51-53
Services, Sun OTP, 7-9
Setting up the provisioning server, Sun OTP, 60-61
SNMP, 18
Solaris OS installation, bootable DVD, 58-59
Solution steps, Sun OTP, 31
Solution topologies, Sun OTP, 13-15

SSH/SSL, 18

Sun OTP

CLI installation, 77-80
component versions, 10-11
components, 23-25
control points, 48
customization, 30-31
deployment methods, 49-50
end-to-end solution deployment, 47-50
GUI installation, 61-76
guides, 11
hardening, 56
host preparation, 59-60
installation log files, 87-88
interfaces, 17-18
order of plan installation, 26
plan settings, 81-85
plans, 25-30
plug-in, 30
product mapping, 10-11
protocols, 18-19
securing web applications, 51-53
services, 7-9
setting up the provisioning server, 60-61
solution steps, 31
solution topologies, 13-15
tools for application development, 21
utility plans, 30
variable set, 27-30
worksheet, 85-86
Sun OTP installation
prerequisite tasks, 58-61
prerequisites, 57-61
Sun OTP Web SSO, integrating web applications, 54-56

T

TCP/IP, 18
Tools for application development, Sun OTP, 21

U

Using N1 SPS

JET, 46

JumpStart, 45-46

Utility plans, Sun OTP, 30

V

Variable set, Sun OTP, 27-30

W

Worksheet, Sun OTP, 85-86

