N1 Service Provisioning System 4.1 User's Guide

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

Part No: 817–4822
February 2004

031219@7518

Contents

Preface

The *N1 Service Provisioning System 4.1 User's Guide* introduces you to the product and describes how to perform many tasks using the HTML user interface.

Who Should Use This Book

This book is for advanced system administrators who are performing the initial configuration of the N1 Service Provisioning System software and for system administrators who are responsible for maintaining the system.

This document is intended for experienced system administrators with extensive knowledge of Sun™ software and hardware. Do not use this document as a planning or pre-sales guide. You should have already determined your system requirements and purchased the appropriate equipment and software before reading this document.

How This Book Is Organized

The *N1 Service Provisioning System 4.1 User's Guide* provides an overview of the N1 Service Provisioning System software and provides task-based procedures for common tasks.

Chapter 1 provides an introduction to the N1 Service Provisioning System software . The product architecture, object model, and interfaces are described.

Chapter 2 prompts you with questions that will aid you in setting up the N1 Service Provisioning System software .

Chapter 3 provides procedures to perform tasks in the N1 Service Provisioning System software ’s HTML user interface.

Chapter 4 provides a description of how the provisioning software interacts with hosts. The chapter also contains procedures on managing hosts with the provisioning software.

Chapter 5 provides conceptual information about what components are, how they work, and which components are built-in. This chapter also provides procedures for basic component management tasks.

Chapter 6 describes how to work with plans in the N1 Service Provisioning System software .

Chapter 7 describes the types of comparisons that the provisioning software can run.

Chapter 8 provides information on how to set up the N1 Service Provisioning System software to notify you when an event occurs.

Appendix A contains a list of names that are restricted to system use only.

Glossary provides a list of words and phrases found in this book and their definitions.

Related Books

The following books provide conceptual information or procedures to administer the N1 Service Provisioning System software . If you plan to use this documentation in a hardcopy format, ensure that you have the following books available for your reference.

- *N1 Service Provisioning System 4.1 Installation Guide*
- *N1 Service Provisioning System 4.1 Reference Guide*
- *N1 Service Provisioning System 4.1 Release Notes*

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see "Buy printed documentation" at `http://docs.sun.com`.

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

CHAPTER 1

An Overview of the N1 Service Provisioning System Software

This chapter introduces the N1™ Service Provisioning System software solution for managing applications in data centers. It discusses the problems that the provisioning software solves. It also describes the provisioning software architecture and the object-oriented methodology that is applied to application management.

This section discusses the following topics:

The Challenges Facing Data Centers

Businesses of all kinds depend increasingly on software applications for their core operations. Managing these applications is a mission-critical task. Yet, until now, IT operators in data centers have had only rudimentary tools for deploying, configuring, and analyzing applications. Most data centers find themselves relying on custom scripts to perform essential functions. IT operators recognize the risk involved with these scripts:

- Scripts are usually hastily written and are prone to error.
- Scripts tend to operate using a long lists of files and lack the ability to manage applications as discrete units.
- Scripts are written with an imperfect knowledge of application requirements, component dependencies, host environments, and other factors that can affect the success or failure of a deployment.

- Scripts lack the benefits of a complete management platform; for example, they typically can not rollback operations systematically, dynamically adjust configuration values, and so on.

The N1 Service Provisioning System Software Solution

N1 Service Provisioning System software is an enterprise-class software platform that automates the deployment, configuration, and analysis of applications in data centers. The provisioning software applies an object-oriented approach to:

- Application components
- Tasks that IT operators perform on application components: deployment, configuration, and analysis.

This object-oriented approach ensures that all the intelligence about an application component is automatically taken into account every time that component is acted upon. This consistency makes data center operations more accurate and less prone to error. Through Application Awareness–knowledge of what an application requires as a whole–IT operators gain unprecedented control over applications and data center operations.

N1 Service Provisioning System software can:

- Automate and manage software rollouts, patches, and upgrades
- Develop models of your existing deployment processes
- Determine what software is installed on your hosts
- Compare the configurations of hosts
- Monitor and maintain documented and consistent configurations.

N1 Service Provisioning System software offers a full range of features that help make that task of managing an enterprise wide computing environment easier and faster.

The provisioning software provides:

Automated Deployment	End-to-end automation of the deployment process, including distribution, configuration, and startup of packaged and custom applications.
Differential Deployment	Enabling delta-only distribution of large content directories thereby speeding up deployments significantly and optimizing incremental directory updates.
Deployment Simulation	Complete simulation of deployments to ensure that key requirements for success are in place before deployment.

Dynamic Configuration	Real-time generation of application configuration for the target environment provides flexibility during deployment.
Dependency Management	Ability to encode application dependency information which is checked during Deployment Simulation to prevent errors that cause downtime and to leverages best practices across entire operations team.
Application Comparison	Ability to track application configuration drifts, and pinpoint unauthorized changes to servers reduces the time required for root-cause analysis of application errors.
Version Control	Central repository that tracks all deployment and configuration data for reference, reconstruction, and to automate rollback to previous states.
Logging and Reporting	Detailed logs of every action taken by the system across all applications and managed servers provides complete audit history of every change made to every server.
Native Windows Support	For COM+ components, no XML editing is required in order to install COM+ components as either InteractiveUser or a particular User and Password.
Windows System Reboot	Many applications require that a server be rebooted during or after software installation. Plans include steps that can reboot a Microsoft Windows server.

The N1 Service Provisioning System Software Architecture

The N1 Service Provisioning System software is a distributed software platform that automates the deployment and configuration tasks in an enterprise wide computing environment and provides increased visibility and control of the servers, installed applications, and file structures.

The provisioning software includes the following special-purpose applications:

- Master Server, a server that hosts the N1 Service Provisioning System software application. This server stores components and plans and provides an interface for managing application deployments. There can only be one Master Server within an enterprise.

- Remote Agent, small management applications that perform operations on the individual host on which it is installed. Every host that is under provisioning software control must have the Remote Agent installed.
- Local Distributors, optional servers that act as a proxy for the Master Server to optimize network communications across data centers, through firewalls, and to reduce the load on the Gold Server.
- Command Line Interface Client, establishes a communication path to the Master Server allowing command execution on the Master Server.
- Web Server, establishes a communication path to the Master Server allowing control of the N1 Service Provisioning System software through the use of a web browser.

The following illustration shows how an example of how applications might be installed on an enterprise network.

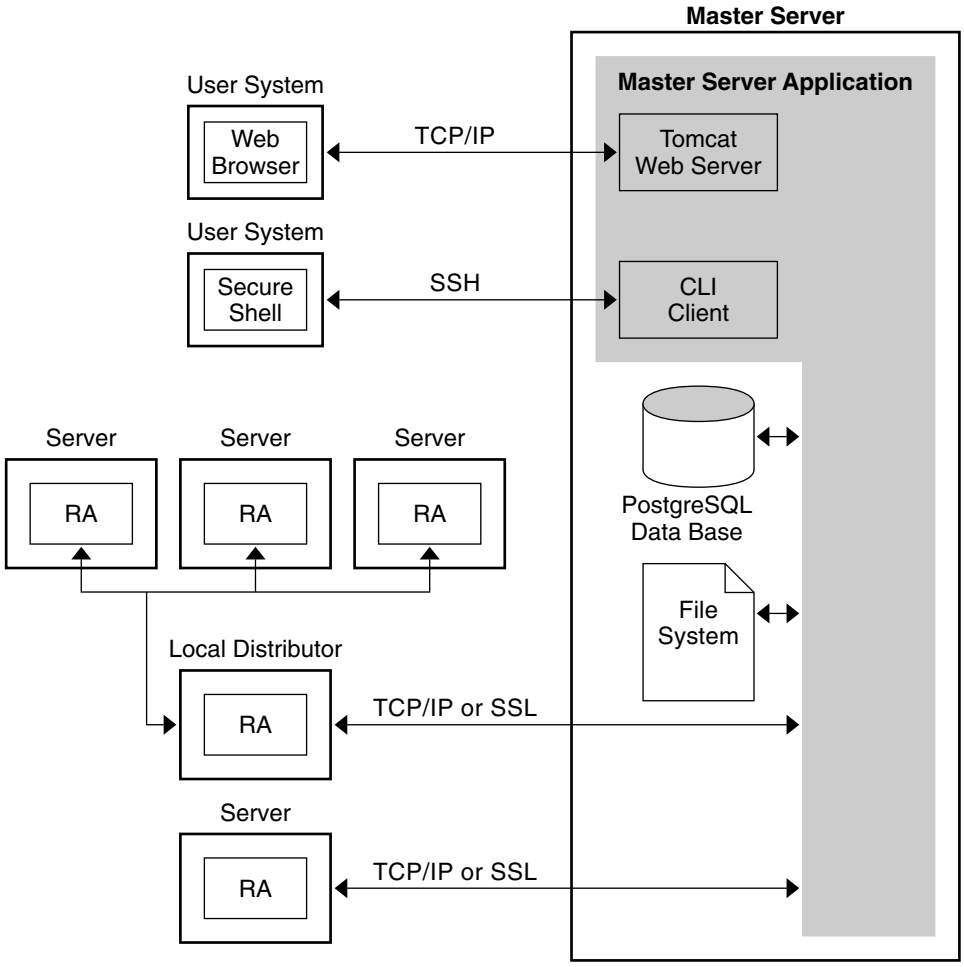

FIGURE 1–1 The N1 Service Provisioning System Software Architecture.

Master Server (MS)

The Master Server is the main processing engine of the N1 Service Provisioning System software . It is installed on a dedicated machine and provides the primary processing engine that drive the various provisioning software functions. The Master Server houses the database that defines all the objects, object attributes, and plans that define the tasks to be performed. The Master Server also runs a Command Line Interface (CLI) client to provide typed control over the N1 Service Provisioning System software and a web server that provides the HTML (graphical) interface.

The Master Server:

- A database identifying all hosts registered in the provisioning software
- A database of system objects, components, and plans
- Performs version control on the objects stored in the repository
- Authenticates IT operators and ensures that only authorized users perform specific operations
- Includes special purpose engines for performing tasks such as dependency tracking and deployments
- Provides both an HTML interface and a command-line interface for users

The N1 Service Provisioning System software repository stores components and plans in a secure, embedded SQL relational database accessible only to authorized users. The repository tracks the version of each component and each plan. For example, as part of a deployment, an IT operator can run plan version 5, which deploys version 3 of a Web server and a version 4 of a custom application.

In live data center operations, proposed changes to applications can come from many sources: from the original application development group, from the QA team, and from the IT team managing production servers. The provisioning software enables IT operators to capture configuration data from any of these sources and check these changes into the repository. IT operators can use the command line interface (CLI) to access any machine on the network and capture its configuration data. Alternatively, they can install a Remote Agent on a machine and then use the HTML interface to select resources from the machine that the provisioning software stores in the repository and combines with configuration data to create a component.

Remote Agent (RA)

A Remote Agent is a Java™ application that runs on every system managed by the N1 Service Provisioning System software . Its job is to perform the tasks requested by the Master Server. Because Remote Agents are typically invoked only when application is being brought up or taken down, Remote Agents do not compete for resources with applications on data center servers.

Remote Agents:

- Report server hardware and software configurations to the Master Server
- Start and stop services
- Manage directory contents and properties
- Caches applications and/or directories and files before actual installation
- Install and uninstall software
- Run OS commands and native scripts specified in component models

Local Distributor (LD)

The use of Local Distributors is optional. When used they become a proxy that temporarily acts as the Master Server to optimizes the distribution and management of applications, files, and directories.

Data centers can use Local Distributors to:

- Minimize network traffic during deployments. The Master Server can send one copy of a component to a Local Distributor, which then replicates the component for installation on a collection of servers through the use of the Remote Agent.
- Minimize firewall reconfigurations. If a firewall stands between the Master Server and a collection of servers, administrators can open the firewall just for the servers running Local Distributors, rather than for every server involved in a deployment.
- Minimize the load to the Master Server during large scale deployments.

Command Line Interface Client

The Command-Line Interface Client provides a communication path to the Master Server to enable the execution of N1 Service Provisioning System software commands from a remote system. These commands are entered using the Windows command line or a UNIX® shell such as bash. The command-line interface also supports the use of shell scripts using sh or Perl.

The Command-Line Interface Client can also use the Jython programming language. Jython is a Java implementation of the high-level, dynamic, object-oriented language Python. You should install Jython on any system on which you plan to install the Command-Line Interface Client. For more information about Jython and to download Jython, visit http://www.jython.org.

Web

The Web provides a communication path to the Master Server.

Network Protocols

N1 Service Provisioning System software supports a variety of network protocols for communication among the N1 Service Provisioning System software applications. The protocols are:

- Raw TCP/IP
- Secure Shell (SSH v1 and v2)

- Secure Sockets Layer (SSL)

Raw TCP/IP

Raw TCP/IP is standard TCP/IP without additional encryption or authentication. The advantage of raw TCP/IP is that it requires no additional set-up and configuration. If your data center network is protected by a firewall and secured from intrusion, using raw TCP/IP provides a convenient method for communication among N1 Service Provisioning System software applications.

Secure Shell

ssh (Secure Shell) is a UNIX-based command suite and protocol for securely accessing a remote computer. `ssh` secures network client/server communications by authenticating both endpoints with a digital certificate and by encrypting passwords. `ssh` uses RSA public key cryptography to manage connections and authentication. Because it is more secure than `telnet` or other shell-based communication methods, many system administrators use ssh to manage Web servers and other remote systems.

The provisioning software can be configured so that its applications communicate using `ssh`. N1 Service Provisioning System software supports OpenSSH explicitly. OpenSSH is a free version of ssh that has been primarily developed by the OpenBSD Project. (For more details, see `www.openssh.com`.) The provisioning software can be configured to support other versions of `ssh`, as well.

Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol for securing communication over IP networks. SSL uses TCP/IP sockets technology to exchange messages between a client and a server, while protecting the message with a public-and-private key encryption system developed by RSA. Support for SSL is included in most Web server products, as well as in the Netscape and Microsoft Web browsers.

N1 Service Provisioning System software applications can be configured to use SSL for their network communications, preventing the provisioning software's messages from being read or tampered with. Optionally, N1 Service Provisioning System software applications can be configured to use SSL to authenticate each other before communicating, further increasing network security.

Selecting Protocols to Meet Specific Needs

N1 Service Provisioning System software enables you to select the protocol you will apply to each of the following types of network communication:

- Communication between the Master Server and its children (Local Distributors and Remote Agents)
- Communication between a particular Local Distributor and its children (Remote Agents)
- Communication between the Master Server and a Command Line Interface Client

You can tailor your network security to meet the needs of your particular network topology. For example, if communication within each of your data centers is secure, but your network connection to a remote data center passes through the public Internet, you could configure the Master Server to use SSL when communicating a Local Distributor installed inside the firewall for the remote data center, so that all communication over the Internet is secured. The Local Distributor could use raw TCP/IP to communicate with its children, since all the communication over the local network is secure, and SSL is not required.

For information on configuring SSL and SSH, please see *N1 Service Provisioning System 4.1 Installation Guide*.

Complex Heterogeneous Data Center Environments

The N1 Service Provisioning System software is designed to fit into data center environments and complement the management, monitoring, and control systems already in place.

Recognizing the diversity of hardware and software found in most Internet data centers, the provisioning software has been designed with cross-platform support in mind. It uses standard communication protocols (HTTP, HTTPS, SSH, and TCP/IP) and standard file and presentation formats (HTML and XML), and it works with standard application architectures (J2EE™ and .Net). It provides data centers with a standards-based system for managing all their applications, whether those applications are UNIX-based or Windows-based.

Supported Operating Systems

You can install the N1 Service Provisioning System software Master Server on systems that are running the following operating systems:

- Solaris 8, Solaris 9

- Red Hat Linux 7.2, 7.3, 8.0 and Red Hat Advanced Server 2.1
- Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server

You can install the N1 Service Provisioning System software Remote Agent, Local Distributor, and CLI Client on systems that are running the following operating systems:

- Solaris 2.6, Solaris 7, Solaris 8, Solaris 9
- Red Hat Linux 7.2, 7.3, 8.0 and Red Hat Advanced Server 2.1
- IBM AIX 4.3.3, 5.1, 5.2
- Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server

For more information about system requirements, see the *N1 Service Provisioning System 4.1 Installation Guide*.

Supported Web Browsers

The following table summarizes the Web browser requirements for the HTML user interface.

TABLE 1–1 Web Browser Requirements for the HTML User Interface

Platform	Browser
Solaris	Netscape 6.2.2, Netscape 7.0
Red Hat	Netscape 6, Netscape 7.1
Windows	Internet Explorer 5.5 and 6, Netscape 6, Netscape 7.1

Supported Locales

The N1 Service Provisioning System software has been internationalized to install and run in localized environments. You will need to adhere to the following requirements if you want to run the software in a localized environment.

- All applications must be run in the same locale or in locales that are equivalent. The Remote Agent, Local Distributors, and CLI Client must run in the same locale as the Master Server.
- The software accepts only ASCII characters for file names, directory names, and other input.

The N1 Service Provisioning System Software Object Model

N1 Service Provisioning System software provides an object-oriented methodology for managing application deployment and configuration. It provides increased visibility and control of the servers, installed applications, and file structures.

The objects are loosely grouped as infrastructure, main, and supporting. Infrastructure objects are things like Users, User Groups, Hosts and Host Sets. Main objects are things like Resources, Components, and plans. Supporting objects are things like Comparisons and Notifications.

Components

A component is a logical grouping of source information (software and/or file structures or other components) that define an application.

The N1 Service Provisioning System software supports two types of components, simple and composite. A simple component is a component that references a single source item. The type of source items that simple component references corresponds to the component's Component Type. For example, a file, a directory of files, a registry key, a COM object, a Java™ Archive (JAR), an EAR, an IIS website, etc. A composite component is a component that references other components. Composite components can contain any number of simple and/or composite components.

All components, regardless of whether they are simple or composite have the following characteristics in common:

Name	A text field that identifies the component. This includes the path where the component is stored.
Type	A user definable object (Component Type) that is used to control how to handle source information. The component type object is actually another component that manages the acquisition and deployment of source objects such as files, directories, and configurations. The provisioning software comes populated with a large number of component types that support WebLogic and Windows, UNIX® , and some generic models.
Version	The revision number of the component. Each time a component is modified, the repository increments its version number.
Platform	Identifies the platform(s) or operating system(s) that are valid targets for the component's deployment.

Source	Identifies from where the component source information came. This include the path.
Checked in	The date and time when the component was checked in. That is, created or modified.
Checked in by	The user ID of the person who checked in the component. This provides an audit trail when trying to troubleshoot problems or inconsistencies.
Label	A user definable text string that can be used to control the sorting on the components page.
Description	A text string that describes the component object. This attribute is not used by the provisioning software but can provide meaningful information to the user.

For a more complete description of component attributes see "Building a Component" on page 94.

The N1 Service Provisioning System software stores each component along with metadata that contains important information about the component, including how to:

- Install and uninstall the component
- Configure the component
- Control the component (e.g., through administration consoles or other interfaces)
- Analyze the component in comparison to other application components (for example, specifying whether log files and /tmp directories should be considered in comparisons)
- Start up and shut down the component

By reading this metadata and comparing components, the provisioning software can identify and track dependencies among components.

Figure 1–2 shows the contents of a component.

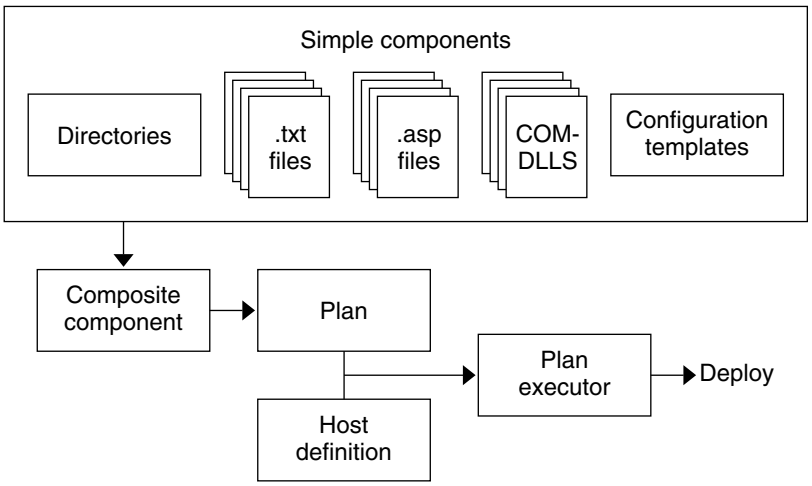

FIGURE 1–2 N1 Service Provisioning System software components include software, templates, and metadata.

In addition to this metadata, components include configuration templates that enable IT operators to adjust specific configuration parameters—for example, port numbers—on a server-by-server basis or according to the rules defined in an execution plan. These variables can be set when an operation is performed, rather than being hard-coded in traditional data center scripts.

Component models are written in XML. The object model incorporates features from the Common Information Model (CIM) and from JSR-77, a Java Specification Request proposing a standard management model for J2EE components. Through the N1 Service Provisioning System software console, you can use component templates to quickly customize existing components. You can also author new templates for applications developed in-house. To perform advanced customizations on complex J2EE components, you can edit components through the HTML interface on the console or with an XML editing tool such as XML Spy.

The provisioning software stores components and plans in a secure repository. Using the console, you check components and plans in and out of the repository, run comparisons, and make changes.

To perform a data center operation such as a deployment, you apply plans to components. Through the N1 Service Provisioning System software , every data center operation incorporates all the knowledge available about each component. Errors are automatically detected. Missing information is flagged. Through automation, deployments and configuration changes become faster and more accurate.

The Component Library

N1 Service Provisioning System software makes it easy for IT operators to begin using component models right away. The Master Server includes a component library with templates for the most common components used in Internet data centers. Table 1–2 lists the templates included in the library.

TABLE 1–2 Templates in the Component Library

Application Component Templates	Simple Files and Directories J2EE Enterprise Archive (EAR) J2EE Web Archive (WAR) J2EE Java Archive (JAR) Solaris Package Solaris OS Patch IIS Web Site or Virtual Directory COM+ Application COM Component MSI Application Windows Registry Keys Windows Data Source .Net Application ASP .Net Application
Web Server Templates	Sun™ONE /iPlanet™ (admin with managed instances) Apache Web Server Microsoft IIS (with Metabase settings)
Application Server Templates	BEA WebLogic (admin with simple, managed, and clustered instances) IBM WebSphere (admin with simple and cloned instances) Microsoft Component Services/COM+ Microsoft Transaction Server (MTS)

In addition to working with these components, the provisioning software can coordinate activities with standard databases, load balancers, and operating systems as part of deploying, configuring, and analyzing applications. Table 1–3 lists the databases, load balancers, and operating systems that can be acted upon.

TABLE 1–3 Databases, Operating Systems, and Load Balancers that can be acted upon

Databases	Oracle Enterprise Microsoft SQL Server
Supported Operating Systems	Solaris™ 6, Solaris 7, or Solaris 8 releases IBM AIX 4.3.x, 5.1, 5.2 Red Hat Linux 7.2, 7.3, 8.0 Red Hat Advanced Server 2.1 Microsoft Windows Server 2000
Load Balancers	Nortel/Alteo WebSystems ACEdirector Foundry ServerIron

Plans

Just as it captures information about applications in component models, the N1 Service Provisioning System software stores information about data center procedures in plans. A plan is an XML document containing a sequence of instructions used to manipulate one or more components or calls to other plans. Plans can encode this information explicitly, or they can leave some details, such as specific server names and port numbers, for the IT operator to specify at run time. When the provisioning software runs a plan, it reads component models for details about configuration requirements, dependencies, and instructions for performing specific operations, such as installing a particular set of files and setting permissions on a directory.

N1 Service Provisioning System software uses plans to:

- Perform operations such as deployments
- Coordinate activities among components during these operations
- Coordinate activities among servers during these operations

There are two types of Plans, simple and composite. Simple plans contain a sequence of instruction that is performed on the target host or hosts unilaterally and cannot call other plans. Composite Plans can only call other Plans so that each step in a multi-step deployment—shutting down an application, removing servers from a load balancer, etc.—can be managed by its own carefully written plan.

For example, a plan to roll out a multi-tiered application might consist of four sub-plans:

- a subplan for removing servers from the load balancer
- a subplan for updating web servers
- a subplan for updating application servers
- a subplan for re-inserting the servers into the load balancer

The rollout plan itself would coordinate activities among the sub-plans and servers. For example, running this plan on a cluster of 10 servers, the N1 Service Provisioning System software would ensure that each subplan had completed successfully and all ten servers were synchronized before proceeding with the next subplan.

Inserting variables in components, plans, and sub-plans gives operators fine-grained control over data center operations. For example, a component can be configured to include a variable for a port number. A plan can assign a value for this variable. Alternatively, the IT operator can assign a value for variable (overriding the value specified in the plan) through the CLI or the HTML interface. Variable settings can be applied to sets of hosts or individual hosts.

By providing a common format for execution plans, the provisioning software replaces the chaotic variety of script formats and languages found in most data centers. Rather than trying to understand the contents of a hastily written script, operators can select the plans they need from a central, version-controlled repository.

Built-In Procedures vs. Plans

The N1 Service Provisioning System software automatically generates procedures for many operations, such as installation and un-installations, for the most common types of components. If your deployments involve installing one component at a time on a single set of hosts, you can use these built-in procedures and never have to author a plan.

Use plans, rather than built-in procedures, if you want to:

- automate the installation of multiple components through a single procedure
- automate the installation of components onto different sets of target hosts
- add dependency checking to installation procedures (e.g., have the installation confirm that a particular component is installed and running before installing the next component)
- synchronize the steps in an installation, so that a particular step does not begin until a previous step has completed successfully

Hosts

The N1 Service Provisioning System software makes it easier to manage servers or hosts by modeling them as objects and storing the model in the host database. The provisioning software enables you to define host types, so you can classify hosts by their configuration or function. The software also enables you to create host sets–groups of hosts that you can manage as a single unit. Host searches enable you to dynamically group hosts based on current attributes. These host management functions, combined with the component and plan objects described above, significantly reduce the complexity of deploying, configuring, and analyzing

applications in data centers. To enable the provisioning software to work with a host it must have a Remote Agent installed, registered in the database, and prepared by the provisioning software for use. Hosts that have gone through all of these processes are also called Nodes. For more information on Hosts see Chapter 4.

Object Attributes

All of the N1 Service Provisioning System software objects have many attributes that either serve to identify the object such as its name, to classify it in some way, provide revision tracking, define grouping, hiding, and so on. Some attributes are unique to a specific object while others are available for some or all of the objects.

Categories

The N1 Service Provisioning System software includes a built-in category called `system`, which is automatically applied to internal resources. All other categories are user definable. As with naming conventions, careful planning should go into defining categories. When category naming conventions are well thought through they become powerful tools when used with searches and views to quickly identify or select groups of objects. Categories for each object are discussed in more detail in the sections relating to each particular object.

You can use categories to classify:

- plans
- components
- resources
- comparisons

Show/Hide

Over time these objects may become obsolete. To reduce the visual clutter you can hide unwanted objects so that they are not shown by default on web pages or listed by CLI commands. It is also possible to delete most of the objects.

Approaches to Modeling

Modeling creates a representation of an application, so that the application can be installed, configured, and managed on targeted host computers. The model for the application can be based on the configuration of a specific host computer running an application, or it can be based on components stored in a repository such as a source code control system.

A host computer that is designated to serve as a reference system for a particular application is called a gold server.[1] In many organizations, a development team, a QA team, or a data center team creates a gold server, so that IT operators have a tested and approved example of an application that needs to be deployed in a production environment.

Using the N1 Service Provisioning System software , you can model an application from a gold server or some other repository. The component model that you create can:

- preserve the unalterable aspects of the configuration–the contents and configuration settings that should be used wherever the application is installed
- establish variables for dynamically adjusting the host- or installation-specific aspects of the configuration (e.g., port numbers or IP addresses that vary from host to host)
- omit the incidental aspects of the configuration (e.g., the contents of log files on a gold server)

Creating models makes applications more manageable. An application model can includes vast amounts of information (files, directories, and so on) that it would be difficult or impossible to manage manually.

A Typical J2EE Modeling Process

In J2EE applications, components typically comprises several resources, such as directories, archives, and so on. In modeling a J2EE application, then, you check in the resources that make up a component and then build the component model based on the resources.

These are the steps typically involved in modeling J2EE applications.

1. Define and configure hosts.
2. Check in resources that make up each component.
   - Install a Remote Agent on the gold server or development system with the components you want to model.
   - Select the resources you want to include in components.
3. Define components.
   - In addition to defining the resources that make up a component, you can create variables for configuration parameters, add information about component dependencies, etc.
4. Write plans for deploying and configuring components.

1 For an overview of the design philosophy behind gold servers, see Traugott and Huddleson, “Bootstrapping an Infrastructure,” Twelfth USENIX Systems Administration Conference (LISA "98), Boston, Massachusetts. This paper, along with others devoted to the issues of IT infrastructure, is available at http://www.infrastructures.org.

5. Run plans.
6. Run comparisons as necessary to analyze your application environment.

A Typical Windows Modeling Process

Windows application components are usually managed as a whole units, rather than as a collections of distinct resources.

These are the steps typically involved in modeling Windows applications.

1. Define and configure hosts.
2. Check in components.
   - Install a Remote Agent on the gold server or development system with the components you want to model.
   - Select the components you want to model. The N1 Service Provisioning System software includes model templates for the most common Windows application components (see Table 1–2 for a list). In many cases, once you have selected the resource template for the Windows component, no further modeling will be necessary.
3. If appropriate, add additional resources to components.
   - In addition to defining the resources that make up a component, you can create variables for configuration parameters, add information about component dependencies, etc.
4. Write plans for deploying and configuring components.
5. Run plans.
6. Run comparisons as necessary to analyze your application environment.

The N1 Service Provisioning System Software Interfaces

There are two interfaces to the N1 Service Provisioning System software :

- an HTML interface accessed through a Netscape™ or Internet Explorer Web browser
- a Command-Line Interface (CLI) accessed through a shell, a Windows prompt, or a script

The HTML Interface

The HTML interface consists of a series of web pages that enable authenticated users to model components, develop and run plans, and perform other operations with the . The HTML interface is designed to be easy to use. It works with both Netscape and Microsoft Web browsers and uses a few basic navigation tools on every page.

All web pages in the HTML interface use the following conventions:

- The left-hand navigation menu appears on all pages (except pop-up windows). The left-hand navigation menu can be collapsed or expanded by clicking on the plus (+) or minus (-) signs.
- All major headings, when clicked, displays a collection of useful links associated with the major heading.
- Information is presented in tables.
- New objects, such as a plans or components, are created by entering information in the top row of a table.
- Clicking Edit takes you to a Web page with graphical controls for editing the content of an object, such as a component.
- Clicking Advanced Edit takes you to a Web page that enables you to edit the XML of a component or plan.
- Information about problems is presented in red.
- The location of the current page is always listed at the top of the page, and the symbol > is used to indicate one page being linked from another (for example, the location Components > Details > Variables Settings tells you that you are viewing the Variable Settings page, which is accessible from the Component > Details page, which in turn is accessible from the main Components page).

The HTML Interface Navigation

Between the many built-in components and the ones you create, the number of components, component types, and plans can become extremely large making it difficult to locate the one you want. To help in locating and using a specific object, easier, the N1 Service Provisioning System software allows you to organize components, component types, and plans in a hierarchical filing system.

The following text describes the Path area of the HTML user interface.

Path	Displays the name of the current working directory.
Show	Allows you to list either components or plans. If you select plans The HTML user interface displays the plans page as though you had click on the plans option in the left-hand navigation menu.
Category	Filters the lists objects by category.

To access objects (either components, component types, or plans) in a specific directory click on the "*change path...*" text. The provisioning software brings up the Change Path page.

Simply enter the desired path into the "selected path" text field or click on the desired icon to select a path.

The Command Line Interface (CLI)

The command-line interface (CLI) is a suite of tools for accessing the provisioning software through non-HTML interfaces, such as a Windows prompt, a shell, or a script. CLI commands can be used in scripts to automate operations, such as checking in files. They can also be used to access the Master Server from systems that lack a Web browser or an HTTP connection.

The CLI is a Command-Line Interface Client which can be installed on any computer that can make a network connection to the Master Server. The client parses the commands into native objects and sends them to the Master Server. Results of the commands are then translated by the client to a textual format and presented to users.

Most CLI commands require authentication. Users authenticate themselves by specifying a user name and a password or a session ID with each command. See "Authentication by Username and Password or Session ID" in *N1 Service Provisioning System 4.1 Reference Guide* for more information.

There are two tools for invoking the CLI:

- `cr_cli`, which operates in single-line command mode, executing one command at a time
- `cr_clij`, which operates in interactive command mode using a Jython interpreter

cr_cli: Single-Line Command Mode

The single-line command mode accepts one command at a time as input. Each command submitted must be complete; the user is not interactively prompted for the next input parameter. Operating in this mode, the Command-Line Interface Client does not maintain a command history.

Here is an example of a CLI command executed with the `cr_cli` tool. This command adds a host of type `prodserver` to the host database. The user running this command, `rbarnes`, supplies his password to authenticate himself to the Master Server.

```
cr_cli -cmd hdb.h.add -u barnes -p bar123 -name webb1  -desc 'web server 1' -tID prodserver
```

`cr_cli` commands can be stored in a file and be called from a shell script. This is useful for repetitive tasks such as running execution plans, comparisons or populating hosts.

cr_clij: Interactive Command Mode

The interactive command line mode uses the Jython interpreter as its shell. Operating in this mode, the CLI offers you these advantages:

- You do not have to type an entire command on a single line. You can simply enter a command name and then enter the command arguments that `cr_clij` prompts you for.
- You can take advantage of the command history stored by the shell.
- You can call a N1 Service Provisioning System software command from within a Jython script.
- You can create more powerful scripts for more complex, repetitive operations.
- For the purposes of automation, the interactive mode results are more detailed.

Note – To call commands from within a Jython script, include the following at the beginning of the script:

```
from clui import *
app=PyCLUI()                make Jython calls to the N1 Service Provisioning System
app.execStr(CLI command)    invoke Jython methods from the
                            new Jython object
App.close()                 delete the instance of this Jython class
```

The Structure of CLI Commands

Whether invoked through cr_cli or `cr_clij`, all CLI commands use the following format:

subsystem.object.command arguments

For example, the command for adding a host to the N1 Service Provisioning System software database is `hdb.h.add`. This command consists three elements that identify:

- a N1 Service Provisioning System software subsystem, in this case the host database (`hdb`)
- an object to operate on, in this case a host (`h`)
- a command or operation, in this case adding (`add`)

Table 1–4 lists the CLI prefix for each N1 Service Provisioning System software subsystem, along with the chapters in this guide that discuss the subsystem.

TABLE 1–4 N1 Service Provisioning System Software Subsystems and Their CLI Prefixes

Subsystem	CLI Prefix	Chapters
Component database	cdb	See Chapter 6, "cdb: CLI Commands for Managing Components" in *N1 Service Provisioning System 4.1 Reference Guide*
Configuration generator	cfg	See Chapter 7, "cfg: CLI Commands for Performing Config-Generation" in *N1 Service Provisioning System 4.1 Reference Guide*
Comparison Engine	cmp	See Chapter 8, "cmp: CLI Commands for Running Comparisons" in *N1 Service Provisioning System 4.1 Reference Guide*
Host Database	hdb	See Chapter 9, "hdb: CLI Commands for Managing Hosts" in *N1 Service Provisioning System 4.1 Reference Guide*
Network Operations	net	See Chapter 10, "net: CLI Commands for Performing Network Operations" in *N1 Service Provisioning System 4.1 Reference Guide*
Plan Database	pdb	See Chapter 11, "pdb: CLI Commands for Managing Plans" in *N1 Service Provisioning System 4.1 Reference Guide*
Plan Execution	pe	See Chapter 12, "pe: CLI Commands for Running Plans" in *N1 Service Provisioning System 4.1 Reference Guide*

TABLE 1–4 N1 Service Provisioning System Software Subsystems and Their CLI Prefixes *(Continued)*

Subsystem	CLI Prefix	Chapters
Resources	cdb.rsrc	See Chapter 13, "CLI Commands for Managing Resources" in *N1 Service Provisioning System 4.1 Reference Guide*
Rules for Notifications	rule	See Chapter 14, "rule: CLI Commands for Notifications" in *N1 Service Provisioning System 4.1 Reference Guide*
User Database	udb	See Chapter 15, "udb: CLI Commands for Managing Users and Groups" in *N1 Service Provisioning System 4.1 Reference Guide*
Categories	cat	See Chapter 16, "Configuration Generation" in *N1 Service Provisioning System 4.1 Reference Guide*

Note – This user manual does not contain a complete list of CLI commands. Each section does list the more frequently used CLI commands that relate to the specific topic. For a complete list of CLI commands, please see *N1 Service Provisioning System 4.1 Reference Guide*.

Using the N1 Service Provisioning System Software

This section provides an overview of how you use the N1 Service Provisioning System software to deploy and configure applications. The steps you will follow vary depending on whether you are using the HTML user interface or the command-line interface. Other chapters in this guide provide detailed instructions for performing each of these steps.

▼ How to use the HTML User Interface

You can use the HTML user interface to model components and to install components on target hosts.

Steps

1. **Install a Remote Agent on each server that you want to model, whether it be a gold server, source code control system, or some other type of server.**

   Note – See the *N1 Service Provisioning System 4.1 Installation Guide* for information on installing Remote Agents.

   ---

2. **Define a host type for the host you will be modeling and for the hosts you will be deploying the application to. Include host type attributes (name-value pairs) for any configuration variables you want to set dynamically.**

   Note – For a description of host types, see "Working With Host Types" on page 76.

   ---

3. **Using the Hosts page, add each host to the host repository. The final step in adding a host is "preparing a host," which enables the provisioning software to fine tune the configuration of the host's Remote Agent in preparation for operating upon the host.**

   Note – For more information about adding and preparing hosts, see "Working with Hosts" on page 61.

   ---

4. **If you are going to define multiple components from a collection of resources, you can check in resources next. Most likely, though, you're working with components that have their own set of resources, so you should begin by creating components directly.**

   - Use the Components page to create a new component.
   - On the Component > Details page, add the resources that make up the component. If a resource has already been checked in, you add it as an existing resource. Otherwise, check in the resource as a new resource.
   - Save the component. Saving the component builds the component–that is, the N1 Service Provisioning System software saves the component with a specific name and version number and associates this specific version of the component with the specific versions of the resources that make it up.

5. **You can now either install the component or to extend the component by editing XML.**

- With most components, you're now ready to run installation procedures using the install procedure that the provisioning software has automatically generated. To run an installation procedure–or some other extended control service associated with this component–display the Component > Details page for the component, select the check box for the procedure you want to run, and click run. This method of running procedures is called direct run.
- If you want to add more information to the component, such as additional configuration variables, additional steps to perform during specific procedures, or more details about what to include and what to exclude during comparisons, check out the component, edit its XML, and then check it back in.

6. **Once you've edited the XML, you're ready to continue with operations.**

   If you want to install one component at a time on a single set of hosts, you can run the install procedure listed on the component details page of the component. If you want to automate the installation of the component on different sets of hosts, add dependency checking to the installation, synchronize steps in the installation, write a plan for the component, and then run the plan to install the component.

▼ How To Use the Command Line Interface

You can also use the command-line interface to model components and install them on target hosts

Steps

1. **Install the Command Line Interface Client on the server where you will be executing CLI commands. The server must be able to establish a network connection to the Master Server.**
2. **Using an XML editor such as TurboXML, write the XML definition of the component.**
3. **Run the `cdb.cd.ci` command to check in the component.**

Note – Once you have checked in the component, you have the option of installing it on a single set of target hosts by using the HTML user interface to directly run the component's install procedure. Direct run procedures can be called through the HTML user interface, but not through the command line interface.

---

4. **Use an XML editor such as TurboXML to write a plan for the component.**
5. **Run the pdb.ci command to check in the plan.**
6. **Run the `pe.p.run` command to run the plan.**

Note – The CLI commands that run plans do not report on the progress of the plan's execution. To monitor the progress of a plan while it is running, run the plan from the HTML user interface.

CHAPTER 2

Assessing Your Application Environment

The N1 Service Provisioning System software provides a framework for organizing your data center and automating data center operations.

This chapter guides you through a series of questions that will enable you to begin using N1 Service Provisioning System software in your data center.

This section discusses the following topics:

Preparing to Use the Command Line Interface

As described in Chapter 1, you can use the CLI to run individual commands and batch scripts on the Master Server.

To begin make a list of the systems from which you expect to use command-line tools, and install the Command-Line Interface Client on each of these systems.

Consider these questions to help you to decide which systems to include in your list.

- From which server are you likely to execute these commands?
- Is there a command console used by the entire data center?
- Does each IT operator have his or her own system for executing commands?

The table below summarizes the planning steps involved with the Command-Line Interface Client.

TABLE 2–1 Preparing to Use the CLI

Question	Action
If you plan to use the CLI, which servers will you run it on?	Install the Command-Line Interface Client on each of these servers. See the *N1 Service Provisioning System 4.1 Installation Guide*.
Is Jython 2.0 or higher already installed on any of these servers?	If Jython 2.0 or higher is installed on a server, you may choose not to re-install Jython from the installer.

Note – You can access HTML interface from any system that can establish an HTTP connection to the Master Server. You do not need to install any N1 Service Provisioning System software on a local machine in order to access the Master Server.

Preparing to Use the HTML User Interface

As described in Chapter 1, you can use the HTML user interface to model components, to develop plans, to perform direct run procedures such as installations, and to develop and run plans that include complex, multi-step procedures.

Before you begin make a list of the systems on which you will be modeling components.

The table below lists questions that will help you plan your use of the HTML user interface.

TABLE 2–2 Preparing to Use the HTML User Interface

Question	Action
Which servers hold the components that you are going to model with the HTML user interface?	Install a Remote Agent on each of these servers. See the *N1 Service Provisioning System 4.1 Installation Guide*.

TABLE 2–2 Preparing to Use the HTML User Interface *(Continued)*

Question	Action
Which web browser is installed on your local system (the system from which you plan to access web pages)?	Make sure you have one of the web browsers listed Table 1–1

Giving Authorized Users Access to the N1 Service Provisioning System Software

Before you begin consider these questions:

- Which users will be using the provisioning software?
- Do you plan to create an account for each user or create a general user account that users will share?
- Do you want to grant different privileges to different users? Restrict certain users from performing certain operations?

The table below will help you plan your management of users and groups.

TABLE 2–3 Managing Users

Question	Action
Which users will be using the system? Will users need individual accounts or generic accounts that they will share?	Define each user account you plan to use. See "Managing User Accounts and User Groups" on page 52.
Will different groups of users be accessing the provisioning software? Would it be convenient to define user groups with different access privileges, rather than assign privileges on a user-by-user basis?	Define user groups and use the permissions assigned to each group to control permissions for users. (All the users in a group receive the privileges granted to that group.) See "Managing User Accounts and User Groups" on page 52.

Managing Hosts

Your data center may include tens, hundreds, or even thousands of hosts. To prepare to manage these hosts, answer the questions listed in the table below.

The table below lists questions that will help you plan your management of hosts.

TABLE 2–4 Managing Hosts.

Question	Action
How do you classify your hosts? What classifications would make them easier to identify and work with?	For each type of host, define a host type. See "Working With Host Types" on page 76.
What hosts are you working with?	Enter each host in the repository by defining it. See "Working with Hosts" on page 61.
How would you group hosts to make them easier to work with?	For each group, define a host set.[1] You can define host sets based on different types of criteria, such as location, purpose, and configuration. A host can belong to more than one host set. Host sets may overlap and may contain other host sets. See "Working with Host Sets" on page 81.
What query executed in real time would identify a group of hosts needed for a particular operation?	To record a query that you will use to identify hosts, define a host search. See "Working With Host Searches" on page 86.

[1] To define a host set based on a host type, use a host search as described in Chapter 4.

Managing Applications

The table below lists the questions you should ask when developing a strategy for managing applications.

TABLE 2–5 Managing Applications.

Question	Action
For each application you want to manage, determine its source: is it a gold server or a source code control system? Where is it?	Install a Remote Agent on the gold server or the server with source code control system. See the *N1 Service Provisioning System 4.1 Installation Guide*.
What are the components that make up this application?	Using the CLI `cdb.c.ci` and `cdb.rsrc.ci` or the Components pages in the HTML user interface, check each application component into the repository.
Do you want to extend the component with additional information and controls? These can include: • configuration variables • dependencies among components • information about content to include or exclude in comparisons	Use an XML editor to add component schema tags to the component. Then check the extended version of the component back into the repository. See "The Component Schema" on page 98 for an overview of component schema tags. See the *N1 Service Provisioning System 4.1 Reference Guide* for details about component schema tags.
Do you want to install the component on a single set of target hosts? Can this component be installed directly without the need for coordinating activities with other components and hosts?	Install the component using the direct run controls in the HTML user interface.
Do you want to automate the installation of this component on more than one set of hosts? Do you want to manage this installation as part of larger process?	Use an XML editor to author a plan for the component. See Chapter 6 for an overview of plans. See the *N1 Service Provisioning System 4.1 Reference Guide* for details about plan schema tags.
When you perform an operation such as an installation, do you want to notify specific users when procedures complete or if exceptions occur?	Use the `rule.add` CLI command to add a notification rule, to edit the XML tags for the component, or to plan to add a `sendCustomEvent` tag. See Chapter 8.
Do you want to compare one application component to another?	Run a model-to-model (M-M) comparison. See "Model to Model Comparisons" on page 170.
Do you want to compare an application component to an installed instance of the application?	Run a model-to-install (M-I) comparison. See "Model-to-Install Comparisons" on page 171.
Do you want to compare two installed instances of the same component?	Run an install-to-install (I-I) comparison. See "Install-to-Install Comparisons" on page 171.

CHAPTER 3

Common Tasks in the HTML User Interface

This chapter tells you how to log in to and out of the HTML user interface, change your password, create and manage user accounts.

This section discusses the following topics:

Common Tasks Page

The Common Tasks page is intended to allow users to quickly access the more frequently performed tasks from a single starting point. This reduces the number of clicks required to perform routine functions such as deployments, getting status reports, viewing run histories, and creating new components. There is also enhanced Windows and WebLogic support.

Windows

The N1 Service Provisioning System software provides enhanced capabilities in out-of-the-box support for Windows Component Types. The eliminates the need for XML editing to install COM+ components as either Interactive User or a particular User and Password.

WebLogic

The N1 Service Provisioning System software provides enhanced capabilities in out-of-the-box support for BEA WebLogic 6.1 and 7.0 applications. No XML editing is required in order to capture, configure and deploy WebLogic Enterprise, EJB™ and Web Applications.

Users can capture a WebLogic application from a reference server, select precisely how this application should be configured and deploy it to either standalone, managed or clustered WebLogic environments.

Note – Ensure that WebLogic applications are not deployed and managed outside of the provisioning software. WebLogic applications must be managed exclusively with the N1 Service Provisioning System software .

Logging in

▼ How To Log in

Steps

1. **In a Web browser, navigate to the home page of the Master Server.**

Note – The Web address for the Master Server is configured during the installation process. If you don't know the Web address for your master server, consult the person who installed and configured the N1 Service Provisioning System software .

---

2. **Enter your user name in the user name field.**
3. **Enter your password in the password field.**
4. **Click the log in button.**

Note – You must hit the tab key after filling in the password field before you can then hit return key to initiate the log in process.

Changing Your Password

You can change your password from the log in page or on the Users Details page for your account.

▼ How To Change Your Password from the Log In Page

Steps

1. **In the left-hand navigation menu on the log in page, click change password.**
2. **Enter your user name in the user name field.**
3. **Enter your current password in the current password field.**
4. **Enter the new password you would like to use in the new password field.**
5. **Confirm that you have typed the new password correctly by entering it in the confirm new password field.**
6. **Click the change password button.**

▼ How To Change Your Password from the Users Details Page

If you have already logged in to the HTML user interface, you can follow these steps to change your password.

Steps

1. **In the left-hand navigation bar, click users.**
   The HTML user interface displays the Users page, which lists the user accounts defined.
2. **In the table listing user accounts, find the row listing your account. In the action column for your row, click details.**
   The HTML user interface displays the Users Details page for your account.
3. **Enter your new password in the new password field.**
4. **Confirm your new password by entering it in the confirm new password field.**
5. **At the bottom of the Users Details page, click the save button.**

Logging Out

▼ How To Log Out

Steps

- **To log out from any HTML user interface page, click log out in the left-hand navigation menu.**

Managing User Accounts and User Groups

The N1 Service Provisioning System software allows you to create user accounts and to define user groups. Because an individual user's permissions are determined by the permissions granted to his or her group some though must go into how accounts and groups are set up.

When setting up these accounts and groups you must decide if groups of users will share a specific account based on the permissions required or if everyone should have their own account to facilitate better audit trails.

User Groups

A user group is a user definable object that is used to categorize users and define permissions. By carefully planning out the names for user groups and which permissions to grant to each group, you can easily manage individual user permissions by making them part of one or more groups. User groups can include one or more individual users. User groups can also include one or more user groups. This a user group can be a super set of all the user groups included in its member list.

The User Group object has the following attributes.

User Group	A user definable text string that names the user group object.
Description	An optional user defined text string that describes the user group object. This attribute is not used by the provisioning software but can provide meaningful information to the user.

Member of Group. . .	Allows the user to add users and/or user groups to the currently selected user group. This section of the user group page has two pull down menus that all the user to select and which user(s) or which user group(s) to add this the currently selected user group.
Current Member Users	Lists the users that are a member of the currently selected user group.
Permissions of Member Users	Lists the permission granted to the members that are part of the currently selected user group.

User Accounts

The N1 Service Provisioning System software restricts access and provides audit trails through the use of user accounts. All users are required to log in to use the application. The Master Server includes a single default account –admin. All other accounts are user definable

The admin User Account

This default account provides initial access to the provisioning software and is intended for system administrators. The user name for this account is admin and the password is admin.

Once you have logged in as `admin`, you can change the account password and set up user accounts and user groups.

Note – Be sure to change the default password for this `admin` account, in order to keep unauthorized users from accessing it.

Other User Accounts

When setting up user accounts you must decide if everyone is to have their own account or if groups of people will share a specific account. In some organizations it may be simpler to create user accounts based on function and give people log in based on the function they perform. For example, everyone who performs administrative tasks may share the `admin` account. Providing every user with a unique account will allow for better audit tracking because the provisioning software maintains a history of what is done and by whom.

The User Group object has the following attributes.

User	A user definable text string that names the user account.
Password section	This section shows wether a password is set or not and allows a password to be set or changed.
Member of User Groups	Allows the user to add or remove the currently selected user account to or from one or more user groups.
Net Permissions for User	Displays the permissions granted to the currently selected user account. These permissions are granted by virtue of which user group or groups to which the user is a member.
Other	Allows the user to deactivate and hide a user account. Because it is not possible to remove user accounts, the provisioning software makes it possible to deactivate and hide obsolete accounts.

▼ How To View User Groups

Steps

1. **In the left-hand navigation menu, click user setup.**

   HTML user interface displays the User Setup page, which provides a link to the User Group page.

2. **In the information area of the page, click user groups. This displays the User Groups page, which lists the user groups already defined. The two arrows in the header of the *user group* column allow you to revers the order of how the user groups names are sorted. The white arrow indicates the order of the sort.**

Note – To view a list of users within a group, find the row listing the group you are interested in, and click details.

▼ How To Create User Groups

Steps

1. **In the left-hand navigation menu, click user setup.**

   The HTML user interface displays the User Setup page, which provides a link to the User Group page.

2. **In the information area of the page, click user groups. This displays the User Groups page, which lists the user groups already defined.**

3. **In the top row of the table listing user groups, enter a name and a description for the new user group, and click create.**

   The HTML user interface displays a User Groups Details page in which you can configure the new user group.

4. **Use the fields on this page to configure the new user group. When you first define a user group, it includes no members.**

   - Add an individual user to the group. In the members of group area of the page, use the pull-down menu in the user row to select the name of an individual user that you want to add to this group. When the menu is displaying the name you want, click add. The HTML user interface updates the User Group Details page, and adds the name of the user to the list of users in the current group members field.
   - Add another user group to this group. In the members of group area of the page, use the pull-down menu in the user group row to select the name of an existing user group that you want to add to this group. When the menu is displaying the name you want, click add. The HTML user interface updates the User Group Details page, and adds the names of all the users in the group you selected to the list of users in the current group members field.

     Note – The current group members field will only list a user once, even if that user belongs to two or more groups that you have added to this group.

   - Set permissions for the new user group. In the permissions of group users area of the page, use the check boxes to define the permissions you want to assign this group. Assigning write permission for an object allows users in this group to enter those objects into the repository. For example, a user with write permission for plans can check plans into the repository. A user with write permission for admin users and groups can create user accounts and define user groups, as you're doing now. You can use the check boxes and menus in the run on host set column to limit the host sets on which the users in this group can run plans and preflights and (independently) the host sets on which they can run comparisons.

5. **When you have finished configuring the group, click save.**

   The HTML user interface saves the configuration you defined and displays the User Groups page, which will now list the new user group.

▼ How To Edit User Groups

Steps 1. **In the left-hand navigation menu, click user setup.**

The HTML user interface displays the User Setup page, which provides a link to the User Group page.

2. **In the information area of the page, click user groups. This displays the User Groups page, which lists the user groups already defined.**

   In the table listing user groups, find the row describing the group you want to modify, and click details.

   The HTML user interface displays a User Groups Details Edit page for the group.

3. **Use the controls on this page to:**

   - Add users or user groups to this user group. In the members of group area of the page, use the pull-down menus and the add links to add users or user groups to this group.
   - Delete users or user groups from this user group. Click remove beside the name of any user or user group you want to remove from the group.
   - Change the permissions of the user group. In the permissions of group users area of the page, use the check boxes to define the permissions you want to assign this group.

4. **When you have made all your modifications, click the save button to save your changes and apply them to the user group.**

▼ How To View User Accounts

Steps

1. **In the left-hand navigation menu, click user setup.**

   HTML user interface displays the User Setup page, which provides a link to the Users and Users Group page.

2. **In the information area of the page, click users. This displays the Users page, which lists the users already defined.**

▼ How To Create User Accounts

Steps

1. **In the left-hand navigation menu, click user setup.**

   The HTML user interface displays the User Setup page, which provides a link to the Users and Users Group page.

2. **In the information area of the page, click users. This displays the Users page, which lists the users already defined.**

3. **In the top row of the table listing user accounts, enter a name for the new user account, and click create.**

   The HTML user interface displays a Users Details page for the new account.

4. **Enter a password for the user account in the new password field, and then enter it exactly the same way in the confirm new password field.**

5. **Use the controls in the member of user groups area to add this user account to one or more user groups.**

Note – In the N1 Service Provisioning System software , permissions are based on user groups, rather than on individual user accounts. By adding this user account to a group, you determine the privileges it will be assigned.

---

6. **If you want to hide this user account, click the checkbox labeled hidden.**

7. **Click save.**

▼ How To Edit User Accounts

Steps

1. **In the left-hand navigation menu, click user setup.**

   The HTML user interface displays the User Setup page, which provides a link to the User Group page.

2. **In the information area of the page, click users. This displays the Users page, which lists the users already defined.**

   In the table listing user users, find the row describing the user you want to modify, and click details.

   The HTML user interface displays a Users Details page for the selected user.

3. **To change the password, enter a new password for the user account in the new password field, and then enter it exactly the same way in the confirm new password field.**

4. **To change membership in any group, use the controls in the member of user groups area to add or remove this user account to one or more user groups.**

5. **To either hide and/or deactivate this user account click the checkbox labeled hidden and/or deactivate.**

6. **Click save.**

Summary of CLI Commands for Account Management

You can use the `udb.g` commands to define, modify, delete, and list user groups.

TABLE 3–1 Summary of User Group (udb.g) Commands

Command Name	Description
udb.g.add	Adds a new user group
udb.g.del	Deletes a user group
udb.g.la	Lists all the user groups
udb.g.lo	Retrieves information about the specified user group.
udb.g.lp	Lists the permissions granted to the specified group
udb.g.lu	Lists the users who are members of the specified group
udb.g.mod	Modifies an existing user group

The `udb.p` commands enable you to display information about the permissions established in the provisioning software.

TABLE 3–2 Summary of Permissions Management (udb.p) Commands

Command	Description
udb.p.la	Lists all permissions.
udb.p.lo	Retrieves the specified permission.

You can use the `udb.u` commands to manage individual user accounts.

TABLE 3–3 Summary of User Account Management (udb.u) Commands

Command Name	Description
udb.u.add	Adds a new user account
udb.u.cp	Changes the password of the specified user
udb.u.la	Lists all user accounts

TABLE 3–3 Summary of User Account Management (udb.u) Commands *(Continued)*

Command Name	Description
udb.u.lo	Retrieves information about the specified user.
udb.u.lp	Lists the permissions granted to the specified user
udb.u.mod	Modifies the specified user account

CHAPTER 4

Hosts

Hosts are physical computer systems connected to your network that provide content to the rest of the network for various purposes.

A host is a machine that serves as either a deployment target, deployment intermediary (Local Distributor), or a deployment source (Gold Server).

There are two types of hosts- Physical and Virtual hosts. This section discusses the following topics:

Working with Hosts

Hosts Introduction

A host is an entity (typically a server) on which you install a component. You manage hosts by defining them. To define a host you must enter them into the host repository (or host database), assign them host types, and specify the configuration variables needed to manage each particular host. Within the N1 Service Provisioning System software there are two major groupings of hosts, physical and virtual.

Physical Hosts

A physical host represents a server in your network. Once the host is defined, information about that host is stored in the host database and can be selected as a target for deployment.

There are two functional roles for physical hosts. A physical host can be configured with a Remote Agent and acted upon directly by a Master Server or a Local Distributor. Additionally, a host can be configured as a Local Distributor and act as a software distributor.

Virtual Hosts

A virtual host is a host that is only a source for deployment, such as an instance of an application server. Multiple virtual hosts can exist on a single physical machine. Every virtual host has a parent host, which can be either a physical host or another virtual host. Virtual hosts can be nested arbitrarily, although the outermost parent host must be a physical host.

It can be helpful to think of virtual hosts as containers for sets of related components.

In this example, there are two physical hosts, Host 1 and Host 2. There are five virtual hosts: Servers 1, 2, 3, and 4, and the Admin Server.

Virtual hosts are useful for maintaining multiple instances of an application on a single physical host. They establish a clear distinction between physical machines and the applications running on them. This distinction is especially useful when managing multiple applications that must interact with one another to function.

If your installation requires you to use an administrative server to install and configure hosts, the N1 Service Provisioning System software allows you to define execution plans that specify a virtual or physical host.

Preparing a Host

Before the N1 Service Provisioning System software can work with a physical host, a Remote Agent must be installed on the physical host and the physical host must be prepared. The provisioning software prepares a host by installing OS-specific utilities on the host's Remote Agent. The HTML user interface gives you the option of preparing host when you save its definition. You can also select hosts listed on the Hosts page and prepare them by running the prepare command. The Hosts page identifies hosts that haven't been prepared, yet, by highlighting them in yellow.

Once a host has been prepared, you can perform operations with it, such as running comparisons and performing installations.

Overview of Managing Hosts

Managing a host involves these steps:

1. Installing a Remote Agent on the host. You can install a Remote Agent either by:
   - Running the installer.
   - Running the Remote Agent Remote Installer script to install a set of Remote Agents.
2. Defining the host. You define a host either by:
   - Entering the host's data on the Hosts > Detail page and saving the host.
   - Running the `hdb.h.add` command.

   Note – Once you have completed these first two steps, you can add the host to a host set or query the host with a host search. However, you cannot run plans or perform comparisons on the host until it is prepared.

   ---

3. Preparing the host. You prepare the host either by:
   - Leaving the "Prepare Host" check box selected when you save the host with the HTML user interface.
   - Selecting the host on the Hosts page and running the "Prepare" command.

   Note – Once you have prepared the host, you can run plans and perform comparisons on it.

Locked and Unlocked Hosts

When the N1 Service Provisioning System software performs an operation on a host, it locks the host, preventing other plans from operating on it at the same time. Locking a host ensures that one operation, such as an installation, does not undo or overwrite the effects of another. It ensures that the provisioning software can accurately determine the status of a host at any time.

The provisioning software manages the lock automatically. You never need to explicitly lock or unlock a host yourself. However, if the provisioning software indicates that a host is locked, you should wait until its status changes before you try to run a new plan on it.

Hosts and IP Addresses

Each host defined in the N1 Service Provisioning System software must have a unique IP address. You cannot define two hosts, even with different names, that share the same IP address. This restriction helps ensure that one user's work, which might be accessing an address through one host name, doesn't interfere with another user's work, which might be using a different host name to access the same address.

The Hosts User Interface

The Hosts Setup Page

The Hosts Setup page allows you to easily access the three most common tasks; creating hosts, creating hosts sets, and creating host searches. You can access these major areas of function from either the main work area of the Hosts Setup page or from the left hand menu bar once Hosts Setup is selected.

The Hosts Page

The *Hosts* page allows you to view a list of the hosts and some basic information about them and to create hosts. The top row of the *Hosts Page* is used to create new hosts. The rows that follow lists the hosts and displays basic information about each host. The *Hosts Page* has the following fields.

(Check Box)	Marks a host for inclusion when one of the actions are clicked in the *Actions for Checked Hosts* area. When you initiate one of the *Actions for Checked Hosts* processes all hosts that are checked will acted upon.
Host	A user definable text string that names the host object. By default, the hosts are listed in alphabetical order by host name. You can change the sort order between ascending and descending by clicking on one of the sort arrows that are in the Hosts header area. The white arrow indicates the sort order. MS, LD, RA, Virtual Identifies the host as either a Master Server (MS), Local Distributor (LD), Resource Agent (RA), or a Virtual Host (Virtual). You can sort the host list by the items in this column by clicking on one of the sort arrows in the header area for this column.
Host Type	A user definable attribute used to categorize hosts. You can sort the host list by the items in this column by clicking on one of the sort arrows in the header area for this column. For more information see "Working With Host Types" on page 76.

OS	Displays the operating system installed on the host. Possible options are any of the supported Windows or UNIX operation systems. This attribute is set at the time the host is created and can not be changed.
Version	Displays the revision level of the host's operating system. This attribute is set at the time the host is created and can not be changed.
Description	An optional user defined text string that describes the host object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other hosts related pages to either create a new host, view more information about a specific host, view the history of the host, or view the installed components for a specific host.
(Colored Bar)	This is a vertical colored bar located to the left of the check box for a given host. It indicates that the host is not prepared. The N1 Service Provisioning System software can not manage a host unless it is first prepared.
Actions for Checked Hosts	From this area you can either prepare or delete the hosts that are checked.

The Host Details Page

The Hosts Details page allows you to view host information and attributes. This page is organized into several general informational and functional sections. The first section addresses the host name, host type, operating system, and attributes. The second section addresses using the Remote Agent on the host. The next section addresses using a host as a Master Server. The next section addresses using a host as a Local Distributor. The next section addresses host relationships. The last section indicates if the host is hidden or not. The Hosts page has the following fields.

General Information

First section of the Hosts Details Page.

Host	A user definable text string that names the host object. This attribute is set at the time the host is created and can be changed using the Hosts Details page by clicking the edit button.
Description	An optional user defined text string that describes the host object. This attribute is not used by the provisioning software, but can

	provide meaningful information to the user. This attribute is set at the time the host is created, and can be changed using the Hosts Details page by clicking the edit button.
OS	Displays the operating system installed on the host. Possible options are any of the supported Windows or UNIX operation systems. This attribute is set at the time the host is created and can not be changed.
OS Version	Displays the revision level of the host's operating system. This attribute is set at the time the host is created and can not be changed.
Locked Status	Indicates whether the host is locked or not. This is controlled by the provisioning software and cannot be changed by the user.
Last Prepared	Displays the date the host was last prepared or that the host is not prepared yet.
Host Type	Displays the host types. For more information on host types see "Working With Host Types" on page 76. For more information on Physical Hosts see "Physical Hosts" on page 62 and for Virtual Hosts see "Virtual Hosts" on page 62.
Attributes	A user-definable object that can control some aspect of a deployment. For example, you can define an attribute calling it port and set its default value to 1020. This can then be used to configure the communication port for an application. This attribute is set at the time the host Type is created. This area contains three fields that displays information about the host attributes. ■ Attribute - Displays the Attribute name. ■ Default Value - Displays the default value for the attribute. ■ Attribute Value - Displays the actual value for the attribute. If this value is different from the default value the background color is different from the rest of the values.

Remote Agent on this Physical Host

This section of the Hosts Details Page displays information regarding the Remote Agent installed on the host.

Connection Type	Displays the communication protocols. For more information, see "Network Protocols" on page 21.

IP Address or Name	Displays the IP Address or IP Host Name for the host. This must match the IP address for the host.
Port	Specifies the user definable port number used for network communication. This must match the port number set during the Remote Agent software installation.
Parent Type	Displays the parent type for the Remote Agent. The Parent Type field is a pull down menu allowing you to choose the Host Type. The options are Master Server and Local Distributor.
Parent Host	Displays the parent host name for the Remote Agent. This defines source host for a deployment.
Advanced Parameters	Displays the XML code used for advance molding. For more information an advanced modelling contact your Sun service provider.
View Connection & Configuration & log...	This is a button that brings up a popup window that displays three separate types of information depending on which button is selected on the popup window. ■ Connection - Displays host connection information. ■ Configuration - Displays host configuration information. ■ Log - Displays host connection log.
Prepare Remote Agent...	This is a button that prepares the Remote Agent on the host. A popup window displays the status of this process while it is running.
Update Remote Agent...	This is a button that updates the Remote Agent on the host. A popup window displays the status of this process while it is running.

This Physical Host is a Master Server

This section displays IP address and port number, if the Master Server software is installed on the host.

IP Address or Name	Displays the IP Address or IP Host Name for the host. This must match the IP address for the host.
Port	Specifies the user definable port number used for network communication. This must match the port number set during the Remote Agent software installation.

Local Distributor on this Physical Host

This section of the Hosts Details Page displays information regarding the Local Distributor if, the software is installed on the host.

Connection Type	Displays the communication protocols. For more information, see "Network Protocols" on page 21.
IP Address or Name	Displays the IP Address or IP Host Name for the host. This must match the IP address for the host.
Port	Specifies the user definable port number used for network communication. This must match the port number set during the Remote Agent software installation.
Parent Type	Displays the parent type for the Remote Agent. The Parent Type field is a pull down menu allowing you to choose the Host Type. The options are Master Server and Local Distributor.
Parent Host	Displays the parent host name for the Remote Agent. This defines source host for a deployment.
Advanced Parameters	Displays the XML code used for advance molding. For more information an advanced modelling contact your Sun service provider.
Host Relationships	This section of the Hosts Details Page displays membership in Host Set(s), and lists any virtual hosts on this host. The following two items relate to this area.
Member of Host Sets	Displays the host sets that this host is a member. There are two links that allow you to either add this host to more host sets or remove this host from host sets.
Virtual Host Children	Lists any virtual hosts on this host.

Other

This is the last section of the Hosts Details Page and allows the user to hide or view a host.

The Host Edit Page

The Hosts Edit page allows you to view and change host information and attributes. This page is organized into several general informational and functional sections. The first section addresses the host name, host type, operating system, and attributes. The second section addresses using the Remote Agent on the host. The next section addresses using a host as a Master Server. The next section addresses using a host as a Local Distributor. The next section addresses host relationships. The next section indicates if the host is hidden or not. The last section indicates if the host is to be updated. The Hosts Edit page has the following fields.

General Information

First section of the Hosts Edit Page.

Host	A user definable text string that names the host object.
Description	An optional user defined text string that describes the host object. This attribute is not used by the provisioning software, but can provide meaningful information to the user.
OS	Displays the operating system installed on the host. Possible options are any of the supported Windows or UNIX operation systems.
OS Version	Displays the revision level of the host's operating system.
Locked Status	Indicates whether the host is locked or not. This is controlled by the provisioning software and cannot be changed by the user.
Last Prepared	Displays the date the host was last prepared or that the host is not prepared yet.
Host Type	A pull down menu the allows you to select one of the user defined host types. For more information on host types see "Working With Host Types" on page 76. Along with one of the user definable hosts type, you must select one of the two radio buttons that indicate whether this is a Physical Host (the default) or a Virtual Host. For more information on Physical Hosts see "Physical Hosts" on page 62, and for Virtual Hosts see "Virtual Hosts" on page 62.
Attributes	A user-definable object that can control some aspect of a deployment. For example, you can define an attribute calling it port and set its default value to 1020. This can then be used to configure the communication port for an application. This attribute is set at the time the host Type is created. This area contains three fields that displays information about the host attributes. ■ Attribute - Displays the Attribute name.

- Default Value - Displays the default value for the attribute.
- Attribute Value - Displays the actual value for the attribute. If this value is different from the default value, the background color is different from the background color of the default values.

Include Remote Agent on this Physical Host

This section of the Hosts Edit Page includes a check box that indicates if the Remote Agent is to be prepared at this time. Once the check box is selected the following items apply. The Remote Agent software must already be installed on the host before it cam be prepared using this section.

Connection Type	A pull down menu to select one of the supported communication protocols. For more information, see "Network Protocols" on page 21.
IP Address or Name	Identifies the IP address to allow a network connection to the host. A host name may also be used if using a host name is supported.
Port	Specifies the user definable port number used for network communication.
Parent Type	A pull down menu that allows you to select the parent type of the deployment source. This is typically either the Master Server or a Local Distributor.
Parent Host	A pull down menu that allows you to select the host name of the source for deployments. The pull down menu lists all the currently defined hosts.
Advanced Parameters	Displays the XML code used for advance modeling. For more information on advanced modeling, contact your Sun service provider.

Include Local Distributor on this Physical Host

This section of the Hosts Edit Page includes a check box that indicates if the Local Distributor is to be prepared at this time. Once the check box is selected the following items apply.

Connection Type	See "*Connection Type*" above.
IP Address or Name	See "*IP Address or Name*" above.
Port	See "*Port*" above.
Test Connection	See "Test *Connection*" above.
Parent Type	See "*Parent Type*" above.

Parent Host	See "*Parent Host*" above.
Advanced Parameters	See "*Advanced Parameters*" above.

This Physical Host is a Master Server

This section displays IP address and port number if the Master Server software is installed on the host.

IP Address or Name	Displays the IP Address or IP Host Name for the host. This must match the IP address for the host.
Port	Specifies the user definable port number used for network communication. This must match the port number set during the Remote Agent software installation.

Host Relationships

This section of the Hosts Edit Page displays membership in Host Set(s) and lists any virtual hosts on this host. The following two items relate to this area.

Member of Host Sets	Displays the host sets that this host is a member. There are two links that allow you to either add this host to more host sets or remove this host from host sets.
Virtual Host Children	Lists any virtual hosts on this host.

Other

The next section of the Hosts Edit Page allows you to hide or view a host.

Options

This is the last section of the Hosts Edit Page and allows you to update the host with the most recent updates.

▼ How To View Hosts

Steps

1. **In the left-hand navigation menu, click hosts Setup.**

   The HTML user interface displays the Hosts Setup page, which allows you to access the three most common tasks; creating hosts, creating hosts sets, and creating host searches.

2. **In the Hosts Setup page, click hosts.**

   The HTML user interface displays the Hosts page, which displays a list of hosts.

▼ How To View a Host's Configuration

Steps

1. **In the left-hand navigation menu, click hosts Setup.**

   The HTML user interface displays the Hosts Setup page, which allows you to access the three most common tasks; creating hosts, creating hosts sets, and creating host searches.

2. **In the Hosts Setup page, click hosts.**

   The HTML user interface displays the Hosts page, which displays a list of hosts.

3. **Find the row listing the host you're interested in, and click details.**

   The HTML user interface displays a Host Details page for the host. The Host Details page presents the configuration details of the host.

▼ How To Create Hosts

Steps

1. **In the left-hand navigation menu, click hosts Setup.**

   The HTML user interface displays the Hosts Setup page, which allows you to access the three most common tasks; creating hosts, creating hosts sets, and creating host searches.

2. **In the Hosts Setup page, click hosts.**

   The HTML user interface displays the Hosts page, which displays a list of hosts.

3. **In the top row of the table listing hosts, do the following:**

   a. **Enter a name for the new host in the Host text box**

   b. **Use the pull-down menu to select a host type for the host**

   c. **Enter a brief description of the new host in the Description text box**

   d. **Click create**

   The HTML user interface displays the Hosts Edit page for the new host.

4. **Use the controls on the Hosts Edit page to configure the new host.**

   a. **In the host type field, use the pull-down menu to select the host type**

b. **Click the appropriate button to indicate whether this is a physical host or a virtual host. For information about physical hosts and virtual hosts, see “Physical Hosts” on page 62 and “Virtual Hosts” on page 62**

c. **If the host type you selected has attributes associated with it, the Hosts Edit page lists the attributes, along with their default values, in the attribute table. To assign a non-default value to an attribute, enter the value in the attribute column for the attribute row describing the attribute**

d. **Click the check box that indicates which type of provisioning software application you want to install on this host**

e. **Configure the connection type, network address, and parent information to correspond to the configuration values you provided the N1 Service Provisioning System software installer when you installed the provisioning software application on this host**

Note – See *N1 Service Provisioning System 4.1 Installation Guide* for information on configuring SSL and SSH network connections.

---

f. **To add this host to a host set, click add to host sets... in the hosts relationships area of the page. In the popup window, select the host set you want to add this host to.**

5. **Click save.**
   The HTML user interface displays the Hosts page, which now lists the new host.

▼ How To Edit Hosts

Steps

1. **In the left-hand navigation menu, click hosts Setup.**
   The HTML user interface displays the Hosts Setup page, which allows you to access the three most common tasks; creating hosts, creating hosts sets, and creating host searches.

2. **In the Hosts Setup page, click hosts.**
   The HTML user interface displays the Hosts page, which displays a list of hosts.

3. **Find the row listing the host you’re interested in, and click details.**
   The HTML user interface displays a Host Details page for the host. The Host Details page presents the configuration details of the host.

4. **On the Hosts Details page click the Edit button located near the bottom of the page.**
5. **Use the controls on the Host Edit page to modify the host's configuration.**
6. **When you have finished your modifications, click save.**

▼ How To Hide a Host

Steps

1. **In the left-hand navigation menu, click hosts Setup.**

   The HTML user interface displays the Hosts Setup page, which allows you to access the three most common tasks; creating hosts, creating hosts sets, and creating host searches.

2. **In the Hosts Setup page, click hosts.**

   The HTML user interface displays the Hosts page, which displays a list of hosts.

3. **Find the row listing the host you're interested in, and click details.**

   The HTML user interface displays a Host Details page for the host. The Host Details page presents the configuration details of the host.

4. **Click the check box labeled hidden at the bottom of the page.**
5. **Click save.**

▼ How To Show a Hidden Host

If the Hosts page is not showing hidden hosts, do the following:

Steps

1. **In the left-hand navigation menu, click hosts Setup.**

   The HTML user interface displays the Hosts Setup page, which allows you to access the three most common tasks; creating hosts, creating hosts sets, and creating host searches.

2. **In the Hosts Setup page, click hosts.**

   The HTML user interface displays the Hosts page, which displays a list of hosts.

3. **In the upper right corner of the Hosts page, click the check box labeled show hidden hosts.**

   The HTML user interface updates the Hosts page, which now lists hidden hosts as well as displayed hosts.

Note – The HTML user interface indicates that a host is hidden by placing a vertical colored bar just to the left of the check box.

4. **Find the row listing the host you want to view, and click details.**

   The HTML user interface displays a Host Details page for the host. The Host Details page presents the configuration details of the host.

5. **To display the host, so that it always shows up, find the row describing the host you want to hide. Click details.**

   The HTML user interface displays a Host Details page for the host.

6. **Click the check box labeled hidden at the bottom of the page, so that the box is no longer selected.**

7. **Click save.**

Summary of CLI Commands for Hosts

The `hdb.h` commands manage target hosts: physical and virtual hosts whose applications you are managing with the provisioning software.

TABLE 4–1 Summary of hdb.h Commands

Command	Description
hdb.h.add	Adds a new host.
hdb.h.del	Deletes a host
hdb.h.la	Lists all hosts.
hdb.h.lo	Retrieves information about a host.
hdb.h.lq	Queries for matching hosts.
hdb.h.mod	Modifies an existing host.

The `net` commands listed below perform networking tasks related to hosts.

TABLE 4–2 Summary of net Commands

Command	Description
net.gencfg	Generates the Transport Config file for a provisioning software application.
net.ping	Checks connectivity to a Remote Agent or Local Distributor by executing the TCP/IP ping command.
net.traceroute	Uses the IP traceroute utility to find the route to a Remote Agent or Distributor

Working With Host Types

A host type is a user definable classification for hosts based on some combination of attributes, such as function, location, or configuration. By assigning each host a host type, you make it easier to group hosts by their most important characteristics. For example, you can define a host type for the hosts that are used as Web servers, another host type for hosts that are used for database servers, and so on. The Web server host type might include attributes such as `Location`, `WebServer_User`, `WebServer_Group`, `HTTP_Port`, and `HTTPS_Port`. The database server host type might include attributes such as `Location`, `DB_Directory`, `DB_InstanceName`, `DB_Port`, and `DB_SecurePort`.

Host types also enable you to set configuration values dynamically when a data center operation is performed. For example, you could define a host type for Web servers, and this host type could include an HTTP port variable. When you run an installation on hosts with this host type, your installation plan could automatically set the HTTP port variable for each host.

Every host you manage with the provisioning software will be assigned a host type–either the default host type, `crhost`, or a host type that you have defined. Each host may only have one host type.

The Default Host Type

The N1 Service Provisioning System software includes a default host type: `crhost`. By default, all hosts are assigned this host type. The `crhost` type includes two attributes:

- host name
- description

There are no default values for these attributes. The values must be specified for each host when you define the host. The `crhost` host type cannot be modified or deleted.

Host Type Attributes

A host type is distinguished by its name and its attributes. An attribute is a name-value pair. Each attribute may have a default value.

For example, you might define a host type, `production_web_hosts`, for production servers running a Web application. The table below describes an example of such a host type.

TABLE 4–3 Host Type Attributes Examples

Attribute Name	Default Value	Comment
name	None.	The name of the host. This attribute is inherited from the `crhost` host type.
description	None.	A description of the host. This attribute is inherited from the `crhost` host type.
location	San Francisco	The name of the data center where the host is located. In this example, an organization has most of its servers in a data center in San Francisco. So, the user has made this the default value.
host_role	web	An identifier for the role this host plays in the Data Center. By default, this variable is assigned the value web, indicating the host acts as a web server.
web_port	80	The number of the port through which users will access this Web application. Because this variable is listed in the host type, it is accessible to plans, command lines, and Web pages and it can be configured dynamically.

Host attributes must begin with a letter and cannot include spaces.

Adding new attributes to the host type results in the attributes and their defaults being added to every host of that type. Removing attributes from a host type permanently deletes those attributes from every host of that type. Attributes cannot be renamed; they can only be deleted and re-added with a new name, which results in the attribute value being reset to the default on all hosts of that type.

When you define a host (an individual target machine), you can selectively override the values of attributes specified by the host's host type. Default values that are not overwritten will be applied to the host.

The Host Types User Interface

The Host Types object is accessed by clicking on the Administration link in the left-hand navigation menu.

The Hosts Types Page

The *Host Types Page* allows you to view a list of host types and create a host type.

You can sort the host types list by clicking on one of the sort arrows in the header area host types column. Click either the ascending or descending arrow to select the desired sort order.

The *Host Types Page* has the following fields.

(Check Box)	Marks a host type for inclusion when host types are deleted.
Host Type	A user definable text string that names the host types object.
Description	An optional user defined text string that describes the host type object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other host type related pages to either create a host type, view more information about a specific host type, or find a specific host type.
Actions for Checked Host Types	Permanently deletes all checked host types.

The Hosts Types Details Page

The *Host Types Details Page* allows you to view and change a specific host type's information and attributes.

The *Host Types Details Page* has the following fields.

Host Type	A user definable text string that names the host types object.
Description	An optional user defined text string that describes the host type object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Attributes	A user-definable object that can control some aspect of a deployment. For example, you can define an attribute calling it port and set its default value to 1020. This can then be used to configure the communication port for an application. This attribute is set at the time the host type is created. This area contains two fields that displays information about the host attributes. ■ Attribute - Displays the Attribute name. ■ Default Value - Displays the default value for the attribute.
Other	Allows you to hide or display the host type.

▼ How To View Host Types List

Steps

1. **In the left-hand navigation menu, click Administrative.**
   The HTML user interface displays the Administrative page.
2. **In the Administrative page, click Host Types.**
   The HTML user interface displays the Host types page.

▼ How To Viewing Host Types Details

Steps

1. **In the left-hand navigation menu, click Administrative.**
   The HTML user interface displays the Administrative page.
2. **In the Administrative page, click Host Types.**
   The HTML user interface displays the Host types page.
3. **Find the row listing the Host type you're interested in, and click details.**
   The HTML user interface displays a Host Types Details page for the Host Type.

▼ How To Create a Host Type

Steps

1. **In the left-hand navigation menu, click Administrative.**
   The HTML user interface displays the Administrative page.
2. **In the Administrative page, click Host Types.**
   The HTML user interface displays the Host types page.
3. **In the top row of the table on the Host Types page, enter a name and brief description for the new host type, and click create.**
   The HTML user interface displays a Host Types Details page for the new host type.
4. **For each attribute you want to assign to the host type, do the following:**
   a. **In the attributes table, enter a name for the attribute in the attribute column.**
   b. **If you would like to assign the attribute a default value, enter a default value for the attribute in the default value column.**
   c. **In the action column, click add.**
5. **If you want to hide the host type, click the check box labeled hidden.**
6. **To save the host type, click save.**

▼ How To Edit a Host Type

Steps

1. **In the left-hand navigation menu, click Administrative.**
   The HTML user interface displays the Administrative page.
2. **In the Administrative page, click Host Types.**
   The HTML user interface displays the Host types page.
3. **Find the row listing the Host type you're interested in, and click details.**
   The HTML user interface displays a Host Types Details page for the Host Type.
4. **For each attribute you want to change, do the following:**
   a. **In the Host Type field, enter a new name.**
   b. **In the Description field, enter a new description.**
   c. **In the attributes table, enter a name for a new attribute in the attribute column.**

d. **If you would like to assign the attribute a default value, enter a default value for the attribute in the default value column.**

e. **In the action column, click add.**

f. **You either move up, move down, or delete existing attributes.**

5. **If you want to hide the host type, click the check box labeled hidden.**

6. **To save the host type, click save.**

Summary of CLI Commands for Host Types

Table 4–4 lists the CLI commands you can use when managing host types. For more information on CLI commands, see *N1 Service Provisioning System 4.1 Reference Guide*.

TABLE 4–4 CLI Commands for Managing Host Types (hdb.ht).

Task	CLI Command
hdb.ht.la	List all host types
hdb.ht.del	Deletes a host type.
hdb.ht.lo	Retrieve information about a host type
hdb.ht.add	Add a host type
hdb.ht.mod	Modify a host type

Working with Host Sets

A host set is a logical grouping of hosts. A host set can include any of the following:

- one or more hosts (referenced by name)
- one or more host searches (which adds all the hosts found by the host searches to the host set)
- one or more other host sets

When you define a host set to include a host search, the host search is performed and the resulting list of hosts determined whenever the provisioning software acts upon the host set. For example, when the provisioning software uses the host set as a target for an installation or a comparison. Host sets based on host searches enable the

provisioning software to group hosts based on their actual configurations at run time. They enable you to be sure you really have found all the hosts that match the search criteria, rather than having to try to remember which hosts are configured a certain way.

Host sets enable you to perform operations on groups of hosts. For example, to install new IIS global filter settings on all the hosts in the Chicago data center, on a host named `barolo2`, and on all the servers running the application SalesExpress, you would define a host set to include:

- the host `barolo2`
- a host search for all the hosts in the Chicago data center (using a host type attribute you have already defined, such as `location`)
- a host search for all the hosts that are running the application SalesExpress

A host can appear in a host set only once. If, in the previous example, the two host searches happened both to retrieve the same hosts (hosts that were both in Chicago and configured to run SalesExpress), each of these hosts would appear in the host set just once.

Platform Host Sets

The N1 Service Provisioning System software includes a hierarchy of nested host sets that are based on the OS platforms of hosts. The top host set is `any`, which includes hosts running any of the supported OS platforms.

The Host Sets User Interface

The Host Sets Page

The Hosts Sets page allows you to view and create host sets. Host Sets are user definable categories that are used to group hosts together. These Host Sets are used with host searches and plans to facilitate identifying and displaying selecting groups of hosts.

You can sort the host sets list by clicking on one of the sort arrows in the header area host sets column. Click either the ascending or descending arrow to select the desired sort order.

The *Host Sets Page* has the following fields.

(Check Box)	Marks a host set for inclusion when host sets are deleted.
Host Set	A user definable text string that names the host set object.

Description	An optional user defined text string that describes the host set object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other host sets related pages to either create a host set, view more information about a specific host set, or find a specific host set.
Actions for Checked Host Types	Permanently deletes all checked host sets.

The Host Sets Detail Page

The *Host Sets Details Page* allows you to view and change a specific Host Sets' information and attributes.

The *Host Sets Details Page* has the following fields.

Top Area	This section of the Host Sets Details page allows you to assign a host set name and description.
Host Set	A user definable text string that names the host set object.
Description	An optional user defined text string that describes the host set object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Member of Set	This section of the Host Sets Details page allows you to assign membership in one or more host sets. The fields in this section are grouped into three columns, Member Type, Name, and action.
Member type	Lists the type of objects that can be included in this host set. The objects included are Host, Hosts Set, and Host Search. ■ Host - Allows you to assign individual hosts to this host set. ■ Host Set - Allows you to make this host set a member of other host sets. ■ Host Search - Displays the Attribute name. ■ Current Member Hosts - Displays a list of the hosts that are a member of this host set.
Other	Allows you to either hide or display the host set.

▼ How To View Host Sets

Steps

1. **In the left-hand navigation menu, click Host Setup.**

   The HTML user interface displays the Host Setup page.

2. **In the Host Setup page, click Host Sets.**

   The HTML user interface displays the Host Sets page. By default, this page lists all the unhidden Host Sets. To include the hidden Host Sets click the checkbox in the upper right hand corner of the page.

3. **Find the row listing the host set you are interested in, and click details.**

   The HTML user interface displays the Host Set Details page for the host set. This page will list the host set's member criteria and the hosts that currently match this criteria and are included in the host set.

▼ How To Create Host Sets

Steps

1. **In the left-hand navigation menu, click Host Setup.**

   The HTML user interface displays the Host Setup page.

2. **In the Host Setup page, click Host Sets.**

   The HTML user interface displays the Host Sets page. By default, this page lists all the unhidden Host Sets. To include the hidden Host Sets click the checkbox in the upper right hand corner of the page.

3. **In the top row of the table listing host sets, enter a name and a brief description for the host set you want to create. In the action column of this row, click create.**

   The HTML user interface displays a Host Set Details page for the new host set. This page includes:

   - a text field listing the name for the host set
   - a text field listing the description for the host set
   - a table labeled members of set which includes controls for defining selection criteria for the host set
   - a display field labeled current member hosts that lists the hosts that meet the criteria of the host set as it is currently defined.

4. **Use the controls on this page to define the host set. Each time you add a host, host set, or host search to the host set, The HTML user interface displays this page and updates the list of hosts in the host set.**

   - To add a host by name, enter its name in the name field, and click add.

- To add a host based on its configuration attributes, click select from list. In the Select Hosts from List popup window, define a host search that selects the hosts you want, and then click return selected hosts to main window.
- To include another host set within this host set, select the other host set's name from the host set pull down menu, and click add set.
- To include the hosts found by a host search, select the host search from the host search pull-down menu, and click add search.
- To highlight the hosts in the host set that match a criterion you have defined, click the highlight link in the row listing that criterion.
- To remove a criterion from the host set configuration, click the remove link in the row listing that criterion.
- To hide the host set, click the check box labeled hidden.

5. **Click the save button at the bottom of the page.**

▼ How To Edit Host Sets

Steps

1. **In the left-hand navigation menu, click Host Setup.**

   The HTML user interface displays the Host Setup page.

2. **In the Host Setup page, click Host Sets.**

   The HTML user interface displays the Host Sets page. By default, this page lists all the unhidden Host Sets. To include the hidden Host Sets click the checkbox in the upper right hand corner of the page.

3. **Find the row listing the host set you are interested in, and click details.**

   The HTML user interface displays a Host Set Details page for the host set. This page will list the host set's member criteria and the hosts that currently match this criteria and are included in the host set.

4. **Use the controls on this page to define the host set. Each time you add a host, host set, or host search to the host set, The HTML user interface displays this page and updates the list of hosts in the host set.**

   - To add a host by name, enter its name in the name field, and click add.
   - To add a host based on its configuration attributes, click select from list. In the Select Hosts from List popup window, define a host search that selects the hosts you want, and then click return selected hosts to main window.
   - To include another host set within this host set, select the other host set's name from the host set pull down menu, and click add set.
   - To include the hosts found by a host search, select the host search from the host search pull-down menu, and click add search.

- To highlight the hosts in the host set that match a criterion you have defined, click the highlight link in the row listing that criterion.
- To remove a criterion from the host set configuration, click the remove link in the row listing that criterion.
- To hide the host set, click the check box labeled hidden.

5. **Click the save button at the bottom of the page.**

Working With Host Searches

A host search is a query run on the repository that yields a list of hosts whose attributes match those specified in the query. You can use host searches to create lists of hosts that have the same host type, are running the same applications, are configured with the same subnet masks, and so on.

In Host Searches are objects with names and attributes. Therefore, when you create a host search, you assign it a name, a description, search parameters, and save it in the repository. Because host searches are objects and saved, you can invoke them later whenever you need to determine which hosts match the criteria described in the host search.

Invoking host searches provides you with up-to-the-moment information about host configurations. Which hosts are configured with a certain host type? With a certain application? Which have applications running on a specified port? Host searches enable to you generate lists of hosts meeting conditions like these.

By defining a host set (a group of hosts) to include the list of hosts returned by a specific host search, you can dynamically generate a list of hosts for use as targets based on the host configurations at run time.

Note – For more information on host sets see "Working with Host Sets" on page 81.

The Host Search User Interface

The Hosts Search Page

The Hosts Search page allows you to view, create, and run host searches.

(Check Box)	Marks a host search for inclusion when host search are deleted.
Host Search	A user definable text string that names the host search object.
Description	An optional user defined text string that describes the host search object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other host search related pages to either create a host search, view more information about a specific host search, or find a specific host search.
Actions for Checked Host Types	Permanently deletes all checked host search.

The Hosts Search Details Page

The Hosts Search page allows you to view, create, and run host searches.

Top Area

This section of the Host Searches Details page allows you to edit host search information.

Host Search	A user definable text string that names the host search object.
Description	An optional user defined text string that describes the host search object. This attribute is not used by the provisioning software but can provide meaningful information to the user.

Search for Hosts Meeting ALL of the Following Conditions

This section defines the criteria for the search and displays the result. The search criteria fields are grouped into three columns labeled Attribute, Comparison, and Value.

Attribute	Contains a pull down menu and two other sets of criteria to set the parameters of a search. You can search by any of several attributes such as host name, host type, or domain name.

It is possible to define what to include as well as what to exclude.

Comparison	Allows you to set whether the search is to equal or contain the value set in the Value column.

Value	Allows you to specify wether the attribute either equals or contains a value.
Host	Allows you to limit a search to either physical host or virtual hosts.
Physical Host	Allows you to limit a search to hosts that are Master Servers, Local Distributors, or have a Remote Agent installed.
Current Found Hosts	Lists the result. of a search.
Other	Allows you to either hide or display the host search.

▼ How To View Host Searches

Steps

1. **In the left-hand navigation menu, click Host Setup.**
   The HTML user interface displays the Host Setup page.
2. **In the Host Setup page, click Host Searches.**
   The HTML user interface displays the Host Searches page. By default, this page lists all the unhidden Host Searches. To include the hidden Host Searches click the checkbox in the upper right hand corner of the page.
3. **Find the row listing the host set you are interested in, and click details.**
   The HTML user interface displays a Host Searches Details page.

▼ How To Create Host Searches

Steps

1. **In the left-hand navigation menu, click Host Setup.**
   The HTML user interface displays the Host Setup page.
2. **In the Host Setup page, click Host Searches.**
   The HTML user interface displays the Host Searches page. By default, this page lists all the unhidden Host Searches. To include the hidden Host Searches click the checkbox in the upper right hand corner of the page.
3. **In the top row of the Host Searches table, enter:**
   - a name for the new host search
   - a brief description of the host search
4. **Click create.**

The HTML user interface displays the Host Searches Details page. This page will display the name and description you entered on the Host Searches page.

5. **Use the controls on this page to define your host search.**

   - Use the pull-down menu in the attribute column to select an attribute to be added to the search. This menu will list any attributes you have defined as part of host types, as well as the following standard attributes:

     - host name
     - description
     - host type
     - ip address
     - parent
     - os
     - os version

     Once you have selected an attribute, use the pull-down menu in the comparison column to select either equals or contains.

     In the value column of the table, enter a string that the attribute should equal or contain.

     In the action column of the table, click add to add this attribute-comparison-value criterion to the search.

     You can add as many of these criteria as you like.
   - Use the check boxes labeled physical and virtual to indicate whether the search should look only for physical hosts, only for virtual hosts, or for both.
   - Use the check boxes in the bottom row of the search criteria table to indicate whether the search should be limited to hosts running a specific provisioning software application: the Master Server, a Local Distributor, or a Remote Agent.
   - To view the list of hosts that match the criteria you have defined so far, click refresh list.

6. **When you have defined your host search, click save.**

   The HTML user interface saves the host search and displays the Host Searches page, which now lists the host search.

▼ How To Edit a Host Searches

Steps

1. **In the left-hand navigation menu, click Host Setup.**

   The HTML user interface displays the Host Setup page.

2. **In the Host Setup page, click Host Searches.**

The HTML user interface displays the Host Searches page. By default, this page lists all the unhidden Host Searches. To include the hidden Host Searches, click the checkbox in the upper right hand corner of the page.

3. **Find the row listing the host Search you are interested in, and click details.**

   The HTML user interface displays a Host Searches Details page for the host Search. This page displays the search criteria for the host search.

4. **Use the controls on this page to define the search parameters for the host search.**

   - Use the pull-down menu in the attribute column to select an attribute to be added to the search. This menu will list any attributes you have defined as part of host types, as well as the following standard attributes:
     - host name
     - description
     - host type
     - ip address
     - parent
     - os
     - os version

     Once you have selected an attribute, use the pull-down menu in the comparison column to select either equals or contains.

     In the value column of the table, enter a string that the attribute should equal or contain.

     In the action column of the table, click add to add this attribute-comparison-value criterion to the search.

     You can add as many of these criteria as you like.
   - Use the check boxes labeled physical and virtual to indicate whether the search should look only for physical hosts, only for virtual hosts, or for both.
   - Use the check boxes in the bottom row of the search criteria table to indicate whether the search should be limited to hosts running a specific provisioning software application: the Master Server, a Local Distributor, or a Remote Agent.
   - To view the list of hosts that match the criteria you have defined so far, click refresh listing

5. **When you have defined your host search, click save.**

   The HTML user interface saves the host search and displays the Host Searches page, which now lists the host search.

CHAPTER 5

Components

This chapter discusses the Component object model. It describes the basic concepts of components, the approaches for modeling components, the component schema (the system of XML tags) that the N1 Service Provisioning System software uses for modeling components, and CLI commands that can be used to manage components.

This section discusses the following topics:

Working with Components

Basic Concepts for Components

A Component is a logical grouping of source information (file(s) and/or directory structures or other components) that define an application and a set of instructions specifying how to handle the source information. A component can be a collection of files and directories, an autonomous archive such as a J2EE Enterprise Archive (EAR) or a COM component, a complete application such as BEA WebLogic, or an operating system-level update such as a patch or service pack. Components can also reference other components.

N1 Service Provisioning System software makes components more manageable by:

- Creating component models that include a carefully defined group of software resources for each component, along with information about how the component should be installed, configured, and analyzed

- Storing components in a repository with version control, so that you can retrieve any previous version of a component
- Making components available to plans, which perform data center operations in a step-by-step manner, taking advantage of the knowledge embedded in each component model
- Enabling components to be compared to one another as well as to installations of software

For an overview of how the N1 Service Provisioning System software manages components and plans, please see "The N1 Service Provisioning System Software Object Model" on page 25.

Modeling Components

The N1 Service Provisioning System software offers great flexibility in modeling components. Depending on the application that you are modeling, you may choose to use any of the following approaches:

- Fully automated modeling
- Extending built-in resource types with XML authoring
- Authoring component models in XML

Fully Automated Modeling

You can check in a component from a gold server or source code control system, have the provisioning software automatically generate a component model, and then use the component model for installation, configuration, and comparison procedures. You can do all this without ever viewing or editing the XML contents of the component model yourself.

We recommend this approach for modeling components in the Windows environment. The provisioning software's built-in resource types can be used to automatically model the most common resources that make up Windows application components. The resource type templates even include built-in procedures for basic operations such as installations, so that you can perform basic operations on common types of resources without even writing a plan.

For a complete list of resource types, see "Built-in Components Types" on page 113.

Extending Built-in Resource Types with XML Authoring

You can begin with an automatically generated component model, as described above, and then customize the component model by editing XML. You can edit a component model's XML by using the Advanced Edit feature on the component's Component > Details page. You can also edit the XML by downloading it to your local system and editing it with an XML-schema-validating editor, such as Turbo XML.

Editing XML enables you to:

- Customize the component with additional variables
- Add calls to extended control services, such as starting and stopping IIS or Windows services

Authoring Component Models in XML

You can write a component model from scratch, using an XML editor and the component schema described in this chapter. Note, though, that you still have to check the component's XML file and the component's resources into the repository in order for the model to be useful.

Overview of Modeling Components

Table 5–1 provides an overview of the steps for modeling components through the HTML user interface and through the CLI. These steps are described in further detail in the "Procedures" on page 96 section of this chapter.

TABLE 5–1 Summary of Steps for Creating Components

HTML User Interface	Command Line Interface
1. Use the Components page to create a new component. 2. On the Component > Details page, add the reference to components that make up the component. If a components has already been checked in, you add it as an existing component. Otherwise, check in the component as a new resource. 3. Save the component. Saving the component creates a new version of the component with a specific name and version number. This new version of the component is associated with the specific versions of the resources that make it up.	1. Run the `cdb.c.lo` command to list the resources that make up the component you want to manage. 2. Using an XML editor such as TurboXML, write the XML definition of the component. 3. Run the `cdb.rsrc.ci` command to check in the resources that make up the component you want to manage.

Building a Component

A key in creating a useful component is building the component.

Building a component selects specific versions of specific resources for use in the component. It assigns a version number to the component, and ensures that version of the component is always associated with specific versions of the component's resources.

Components consist of the following attributes.

- Path - A user definable location to store components. Between the many built-in components and the ones you create, the number of components can become extremely large making it difficult to locate the one you want. To help locating and using components easier, the provisioning software allows you to organize them in a hierarchical filing system.
- Component - (Name) A user definable text string that identifies the component. In some pages the component (name) will include the associated path such as in the Component Details page. In other pages such as the Components page it does not.
- Type - A user definable name for an object (Component Type) that serves as a service that manages the acquisition and deployment of source objects such as files directories, and configurations. The component type object is just another component that is used to control how to handle source information.

  The N1 Service Provisioning System software comes with a large number of component types that support WebLogic and Windows, UNIX, and some generic models.

- Version - The provisioning software manages version control by assigning a new version number each time a component is modified. This version number has a major number—the number to the left of the decimal point—and a minor number—the number to the right of the decimal point. You have the option of incrementing the minor number or the major number.
- Platform - Specifies which operating system(s) are valid for hosts that are targets for deployment.
- Checked in - The date and time the component was checked in. When a component is checked in it time and date stamps the component.
- Checked in by - The provisioning software provides an audit trail by storing the user ID of the person checking in the component.
- Label - A user definable text string that is used to categorize or group components. Labels can be used as sorting criteria on the component page.
- Category - *A user definable object that is used to filter the component list. Categories are created using the categories page accessed from the Administrative page. Categories are then applied to a component from the Components page. Categories can also be created by clicking in the "Apply Categories . . ."* link. at the bottom of the components page.
- Description - An optional user defined text string that describes the component object. This attribute is not used by the provisioning software but can provide meaningful information to the user. The better a description is the more useful it is for the user.
- Source - Identifies where the component source information came from including the path. This may be an object to be deployed (simple component), other components consisting (composite component) each consisting of other components or of an object to be deployed.
- Component Variables - User definable containers that store values required to deploy a component resource. For more information on component variables see "Component Variables" on page 99.
- Procedures - A set of instructions that tells what to with the resources and variables.
- Hidden - A flag that marks a component for exclusion from lists. Hidden components are not included in lists unless specifically requests.

If you're using the HTML user interface to define a component, saving the component automatically builds it.

Once a component has been built, it is ready to be deployed at which point you can either:

- install the component on a single set of hosts, using the HTML user interface to directly run the install procedure
- write a plan to install the component on several sets of target hosts

Plans are described in Chapter 6.

Procedures

Component procedures contain the logic for how to install, uninstall, and control a component. A given component can define any number of procedures. For example, it can define more than one way to install or uninstall a component. An example of a control procedure might be some logic for starting and stopping a server.

The only way to add install, uninstall, and control procedures is through the advanced edit feature in the HTML user interface or through the XML check-in process using the CLI.

When you create a composite component, by default it will inherit a default install procedure and a default uninstall procedure. In general, components often inherit their procedures from their component type. For more details on inheritance see “Component Inheritance” on page 96.

Component Inheritance

Inheritance the characteristic of getting certain attributes from something else. For example, when a component is created, by default, it gets or inherits variables, snapshots, and/or procedures from its component type.

This makes the component model much more powerful and flexible. For example, if you have a thousand components that all extend the IIS Application Component Type and you need to add additional functionality to these components you do not need to add it a thousand times, you only need to add it to the base component type that all these components extend. All one thousand components will inherit the change from the component type.

Access Modes and Modifiers

Access modes and modifiers control how a component attributes can be used and by whom or what. Access modes and modifiers can effect contained components, component variables, procedures, and source data.

There are basically four access modes, public, private, path, and protected. While some access mode apply to all of the attributes others do not. There are two modifiers final and abstract.

Note – Both the concept of inheritance and the access modes and modifiers function like their Java language counterparts. To understand either of these more fully, please refer to the "Plan and Component Language Specification" as well as the any books that explain these concepts for Java.

Component Access Modes and Modifiers

The access attribute for a component controls how a component can be deployed. There are two modes, path and public. If access is set to path, then the component can only be referenced by other components in the same path and cannot be directly installed by the user. If access is set to public, then no such restrictions apply. A component can be referenced by any other component regardless of its path and it can be deployed directly by a user. The default access mode for a component is public.

The modifiers, abstract and final, for a component effect deployment options. If the modifier is set to abstract then the component may not be installed. It serves only as a base component for other components to extend. Only abstract components may declare abstract child elements. If the modifier is set to final then the component may not be extended by another component. If the modifier is omitted, then the component may be extended and installed.

Component Variable Access Modes and Modifiers

The access attribute for a component variable controls what can access the variable. There are four access modes, public, private, path, and protected. If the access mode is set to public, then access to the variable is not restricted in any way. If the access mode is set to protected, then access to the variable is restricted to derived components and entities in the same path. If the access mode is set to path, then access to the variable is restricted to entities in the same path. If the access mode is set to private, then access to the variable is restricted to the component containing the variable. The default access mode for a component is public.

The modifiers, abstract and final, for a component variable effect the default values and overrides for a component variable. If the modifier is set to abstract then the variable default attribute is omitted and must be specified by non-abstract derived components. Variables can only be declared abstract if the component is also declared abstract. Abstract variables may not be private. Non-abstract variables must declare a default value. If the modifier is set to final then the variable may not be overridden by derived components. If the modifier is omitted, then derived components can choose whether or not to override the variable.

Component Source Access Modes and Modifiers

The access attribute for a component source is not supported and therefore is implied to be public.

The modifiers, abstract and final, for a component source effect the overrides for the component source location. If the modifier is set to abstract then the source location is omitted and must be specified by non-abstract derived components. The source location for a component can only be declared abstract if the component is also declared abstract. If the source location is declared non-abstract then the source

location must be specified. If the modifier is set to final then the source location may not be overridden by derived components. If the modifier is omitted, then derived components can choose whether or not to override the location for the source data.

The Component Schema

The <component> Tag

The `<component>` tag is the basic tag for defining a component. The `<component>` names the component, specifies its version number, and includes optional information such as the software vendor that create the application modeled by the component.

Other Component Tags

Within the opening `<component>` tag and the closing `</component>` tag in the component's XML file are a number of other tags that define:

- the list of resources that make up the component
- what to install (by default, all the resources) in the component
- what to remove when un-installing the component (by default, all the resources that were installed)

By editing the XML definition of the component, you can add other tags that define:

- configuration variables for the component
- additional procedures that can be called to manage the component
- content that should be included or excluded when the component is analyzed

```
<component attributes>
    component_contents
</component>
```

Table 5–2 describes the XML tags that you can use to modify or extend a component.

TABLE 5–2 XML Tags for the Component Schema

Schema Tag	Required/Optional	Description
resourceList	Required	A list of the resources that make up the component. If you define the component through the HTML user interface, this list will include each resource that you have added to the component.

TABLE 5–2 XML Tags for the Component Schema *(Continued)*

Schema Tag	Required/Optional	Description
varList	Optional	Contains a list of configuration variables for the component. Each variable is assigned a name and, optionally, a default value.
installList	Required.	Includes one or more installSteps tags, each of which includes a grouped block of installation instructions. By default, installList includes a single installSteps block which includes a single command: deployAllResources.
uninstallList	Required	Includes one or more uinstallSteps tags, each of which includes a grouped block of instructions for un-installing (removing) the component. By default, uninstallList includes a single installSteps block which includes a single command: deployAllResources.
controlList	Optional	Defines one or more higher-level procedures, such as start and stop, that can be called to control the component.
snapShotList	Optional	Defines what should be included in a snapshot. Each snapshot can be assigned a name, a list of files to be included, and operations (such as console commands) to be performed before or after the snapshot is taken.
diff	Optional	Defines what should be excluded during snapshots.

For details about each of these tags, please see *N1 Service Provisioning System 4.1 Reference Guide*.

Component Variables

Components support the use of variables, which are user definable containers that store values used during deployment. The purpose of component variables is to make parts of the component accessible and configurable external to the component. For example, a component may have a variable named `installPath` that is overridden on a per host basis that defines where to install each component.

The value of the component variable can be a reference to other component variables, including variables that are defined by the component's container. When a nested component is added to a container component, the HTML user interface verifies that the referenced variable is defined within the container. If the variable is not defined within the container component, the HTML user interface automatically adds the referenced variable to the container's variable list. As an example, it is typical for simple components to define their `installPath` variable to have the value of the

container component's `installPath` variable. In this example, the syntax of the referenced variable is `:[container:installPath]`. For more information on variable substitution, see Chapter 16, "Configuration Generation" in *N1 Service Provisioning System 4.1 Reference Guide*.

Component variables are evaluated and assigned a value at the time the component is deployed. They can be used to define host names, IP addresses, paths, or any other piece of information that is required to implement a deployment.

Variable Sets

A variable set is a list of component variables with alternate values for one or more of the variables overriding their default value. This allows the user to specify different values for component variables based on which variable set is used.

An example of using a variable set is when a component is deployed into different environments such as a production environment and a development environment. If the variables defined in a component allow for the differences between both environments then you can use different variable sets based on where the component will be installed. At plan run time, the user simply chooses which variable set to use based on which environment the component will to be installed.

Variable Overrides

Variable overrides only pertain to composite components and are not used with simple components.

When a component contains other components (children components), a variable set only effects the top-level parent component. All children components use the default values for their variables.

The way to override the defaults values for children components is to set the variable override variables when creating a component that contains children components. Each referenced component will have a set of variable overrides that you can use.

Component Reports

The N1 Service Provisioning System software provides a number of lists that provide useful information about components. It is possible to list all the places where a specific component has been installed, the resources included in a specific component, all the versions of a specific component (the component's version history).

The Components User Interface

The Components Page

The Components page allows you to list components, view component details, create components, and edit components. This page also contains controls for filtering the component list and selecting sort criteria. You can change the sort criteria for the components list by clicking on one of the sort arrows in the header area. Click either the ascending or descending arrow selects that column for sorting and the sort order.

Path	Allows you to specify where to look for or store components. Click on *change path* to specify a different path.
Show	Allows you to list either components or plans. If you select plans the HTML user interface displays the plans page as though you had click on the plans option in the left-hand navigation menu. You can also narrow the list by selecting a category from the category pull down menu.
(Check Box)	Marks a component for inclusion when one of the actions are clicked in the *Actions for Checked components* area. When clicked, *Actions for Checked components* will either delete, check in, or categorize a component.
Component	A user definable text string that names the component object. By default, the components are listed in alphabetical order by component name. You can sort the components list by the items in this column by clicking on one of the sort arrows in the header area for this column. Click either the ascending or descending arrow selects this column for sorting and the sort order.
Type	A user definable object that is used to control how to handle resources. For more information on component types see “Building a Component” on page 94.
Version	Displays the revision number of the component. Each time a component is modified, its version number is incremented.

You can sort the components list by the items in this column by clicking on one of the sort arrows in the header area for this column. Click either the ascending or descending arrow selects this column for sorting and the sort order.

Label	A optional user definable text string that can be used to further group components together.

	You can sort the components list by the items in this column by clicking on one of the sort arrows in the header area for this column. Click either the ascending or descending arrow selects this column for sorting and the sort order.
Description	An optional user defined text string that describes the component object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other component related pages to either create a new component, view more information about a specific component, or edit components.
Actions for Checked Component	From this area you can either delete, check in, or apply categories to the checked components.

The Components Details Page

The Component Details page allows you to view Component information and attributes. This page is organized into several general informational and functional sections and also contains buttons that allow you to show where the component is uninstalled, delete the component, go to the Component Edit page, and download XML files.

General Information

This section of the Components Details Page displays general information about the component.

Component	The name of the component. By default, the components are listed in alphabetical order by component name. You can sort the components list by the items in this column by clicking on one of the sort arrows.
Type	The name of the component type. A user definable object that is used to control how to handle resources. For more information on component types see “Building a Component” on page 94.
Version	Displays the revision number of the component.
Platform	Displays the operating system(s) that are valid targets for component deployment.

Checked In	Displays the date and time when the component was checked in. That is, created or modified.
Checked In By	Displays the user ID of the one who checked in the component. That provides an audit trail when trying to troubleshoot problems or inconsistencies.
Label	A user definable text string that can be used to control the sorting on the components page.
Category	A user definable object that is used to filter the component list. Categories are created using the categories page accessed from the Administrative page. Categories are then applied to a component from the Components page. Categories can also be created by clicking in the "*Apply Categories . . .*" link.
Description	An optional user defined text string that describes the component object. This attribute is not used by the provisioning software but can provide meaningful information to the user.

File/Directory Information

This section only appears if the component type is a file and displays general information about the file. The directory section is identical to this one except that it is labeled directory.

File Name	This is a the text string that names the component type. This field is called *Directory Name* for a directory.
Time Stamp	The date and time the file or directory was created.
Total Size	The total amount of disk storage space required to store the file or directory.
User	This is the owner of the file or directory.
Group	This is the group relationship for this file or directory.
Permissions	This shows the Read, Write, and Execute permissions for the User, Group, and everyone else not in the first two categories. The first three characters represent the R (read), W (write), and X (execute) permissions for the user (owner). The second three characters represent the R (read), W (write), and X (execute) permissions for the group. The third set of three characters represent the permissions for everyone else who is not either the owner or part of the group.
Options	Displays the options that were checked when the component was created.

Referenced Components

This section of the Components Details Page displays components that are referenced by the parent component.

Resource [tree]	A user definable text string that names the component type that is a child of the parent component (resource object).
Type	This identifies the component type for the child component.
Version	Displays the revision number of the resource. Each time a resource definition is modified, its version number is incremented.
Source	Identifies the source path for the child component (resource).
Action	Allows you to either check in (define) a new resource for the component or select resources from a list of resources that already exist.

Component Variables

This section of the Components Details Page displays information about the component variables.

Variable	A user definable variable that can be used to customize deployments.
Value	A user specified value for the component variables.

Component Procedures

This section of the Components Details Page displays information about the component procedures.

Checkbox	Marks a procedure for inclusion in a plan.
Procedure	Lists the procedures defined in this component.
Actions	Generates a plan that incudes the checked

Buttons

This section of the Components Details Page displays information about the component procedures.

Edit	Takes you to the component edit page allowing you to change component information.
Advanced Edit	Takes you to the XML edit page allowing you to change component information and function at the code level.

Delete	Allows you to delete this component. If some other object references this component, the HTML user interface displays a message and will not allow you to delete the component.
Done	Takes you back to the components page.
Download	Allows you to export the component XML code to a text file.
Variable Settings	Allows you to override the default value of any variable.
Where Installed	Displays where the component has been installed.

The Components Edit Page

The Component Edit page allows you to change Component information and attributes. However, not all fields can be changed.

As with the details page, the edit page displays different fields depending on which component type is displayed. For example, the fields for a file are different from that of an IIS application. The following only discuss the fields that yo can change.

Also, as with the details page, the edit page is organized into the same several general informational and functional sections as the component details page. Each section will be discussed individually.

General Information

This is the first section of the Components Edit Page and displays general information about the component.

Component	Displays the name for the component object. This field was set when the component was created and cannot be changed.
Type	Displays the name for the component type. This field was set when the component was created and cannot be changed
Version	Displays the revision number of the component. Each time a component is modified, its version number is incremented. The version number cannot be changed by the user.
Platform	Allows you to change the operating system(s) that are valid targets for component deployment. This is done by selecting an item from the pull down menu.
Checked In	Displays the date and time when the component was checked in. This field is set by the provisioning software and cannot be change by the user.

Checked In By	Displays the user ID of the one who checked in the component. This field is controlled by the provisioning software and cannot be change by the user.
Label	A text field that allows you to enter or change the label.
Category	Displays the component category. This field was set when the component was created and cannot be changed.
Description	A text field that allows you to enter or change the component description.

File/Directory Information

This section only appears if the component type is a file and displays general information about the file. The directory section is identical to this one except that it is labeled directory. None of these fields are changeable.

Resources Included

This section of the Components Edit Page displays information about the component's resources. Some fields can be changed and some cannot.

Resource [tree]	Displays the path to the resource component.
Type	Allows you to change the resource component type by selecting a new type form the pull down menu.
Version	Displays the revision number of the resource. This field is controlled by the provisioning software and cannot be changed by the user.
Source	Identifies the source for the resource.
Action	Allows you to either check in (define) a new resource for the component or select resources from a list of resources that already exist.
(Check Box)	Marks a resource for inclusion when one of the actions are clicked in the *Actions for Checked Resources* area. When clicked, *Actions for Checked Resource*s will either check in, delete, move up or down and resource, or set the resource version to the most recent one.

Component Variables

This section of the Components Edit Page displays information about the component's variables.

Variable	This field allows the user to create new variables. This column also lists existing variables.
Value	This field allows the user to set the value for new variables. This column also lists the value for all existing variables.
(Check Box)	Marks a component variable for inclusion when one of the actions are clicked in the *Actions for Checked Variables* area. When clicked, *Actions for Checked Variables* will either move up, move down, or delete variable.

Buttons

This section of the Components Details Page displays information about the component procedures.

Check in	Checks in the changed version of the component and increments the revision number.
Check in as	Checks in the changed version of the component using a different name and sets the appropriate revision number.
Cancel	Takes you to the components details page without saving any changes.

▼ How To View Components List

Steps

- **In the left-hand navigation menu, click components.**
  The HTML user interface displays the components page, which lists the components already checked in to the database.

▼ How To View Components Details

Steps

1. **In the left-hand navigation menu, click components.**
   The HTML user interface displays the components page, which lists the components already checked in to the database.
2. **Find the row listing the component you're interested in, and click details.**
   The HTML user interface displays a Components Details page for the component.

▼ How To View A Component's Installations

Steps

1. **In the left-hand navigation menu, click components.**

   The HTML user interface displays the components page, which lists the components already checked in to the database.

2. **Find the row listing the component you're interested in, and click where installed.**

   The HTML user interface displays a Components Where Installed page for the component.

▼ How To Create Components

Steps

1. **In the left-hand navigation menu, click components.**

   The HTML user interface displays the components page, which lists the components already checked in to the database.

2. **In the top row of the table listing components, enter a name in the component field for the new component, and click create.**

   The HTML user interface displays the Components Details Edit page for the new component. This page is slightly different from the component edit page discussed earlier in that the name and type can be changed here.

3. **Use the controls on this page to specify the necessary information that will make up the component.**

   - Change the name in the component field, if desired. Once the component is checked in you will not be able to change the name.
   - Select the type from the pull down menu. The component edit page fields will change to match those relevant to the component type selected.
   - Select the platform from the pull down menu.
   - Enter a label in the Label field.
   - Enter a description in the Description field.
   - Continue entering the required information. The information required from here will depend on the component type selected.
   - When finished click *check in*.

   The HTML user interface displays a window reporting that it is checking in the component as version 1.0.

4. **Click continue to check in to finish checking in the component.**

▼ How To Edit Components

Steps

1. **In the left-hand navigation menu, click components.**

   The HTML user interface displays the components page, which lists the components already checked in to the database.

2. **Find the row listing the component you're interested in, and click details.**

   The HTML user interface displays a Components Details page for the component.

3. **Click the edit button.**

   The HTML user interface displays a Components Details Edit page for the component.

4. **Use the controls on the Components Edit page to modify the component's configuration.**

5. **When you have finished your modifications, click check in.**

6. **When you have finished listing all the resources for the component, click check in.**

   The HTML user interface displays a window reporting that it is checking in the component as version nth which will be the next incremental number.

7. **Click continue to check in to finish checking in the component.**

Where to Find Information Reported by the N1 Service Provisioning System Software

The following table tells you how to find some of the information reported by provisioning software.

Topic	To Find . . .	Do This . . .
Components	All the places where a specific component has been installed.	■ Navigate to the Components page. ■ Find the row listing the component you want. Click where installed.
	A list of the resources included in a specific component.	■ Navigate to the Components page. ■ Find the row listing the component you want. ■ Click details.
	A list of all the versions of a specific component (the component's version history).	■ Navigate to the Components page. ■ Find the row listing the component you want. ■ Click details. On the Components Details page, click version history.

Summary of Component CLI Commands

The `cdb.c` commands provide general-purpose controls for managing components.

TABLE 5–3 CLI Commands for Managing Components

Command	Description
cdb.c.ci	Checks in non-browsable components and component models.
cdb.c.co	Checks out a component.
cdb.c.la	Lists all versions of all components.
cdb.c.lo	Lists detailed information about a component.
cdb.c.lv	Lists all versions of a component.

TABLE 5–3 CLI Commands for Managing Components *(Continued)*

Command	Description
cdb.c.mod	Modifies a component.
cdb.c.sc	Applies one or more categories to a component.
cdb.c.sh	Shows or hides a component.
cdb.c.del	Deletes a component

The `cdb.ic` commands retrieve information about components that are already installed on hosts.

TABLE 5–4 CLI Commands for Managing Installed Components

Command	Description
cdb.ic.lbc	Lists all the hosts on which a component is installed.
cdb.ic.lbh	Lists all the components installed on a specific host.
cdb.ic.vs.lo	Lists details of the specified generated variable settings object.

The `cdb.vs` commands manage variable settings for components.

TABLE 5–5 CLI Commands for Managing Variable Settings

Command	Description
cdb.vs.add	Adds a new variable settings object.
cdb.vs.del	Deletes a variable settings object.
cdb.vs.imp	Imports a variable settings object from one component into another.
cdb.vs.la	Lists all the variable settings objects associated with a specific component.
cdb.vs.lo	Lists the details of a specific variable settings object.
cdb.vs.mod	Modifies a variable settings object.

Working with Component Types

Introduction

A Component Type is a user definable object that is used to control how to handle source items referenced by a component. The component type object is actually another component that manages the acquisition and deployment of source items such as files, directories, and configurations.

All components must have its component type attribute set to some component type. Even if a component does not have a defined component type, its component type is set to `untyped`.

The files, directories, and other tree structures referenced by a simple component is managed as a discrete unit within a component. For example, an IIS application, which the provisioning software would manage as referenced source items might include the following:

- a directory of content
- IIS Web site settings
- COM+ application
- Windows Registry settings

Some source items referenced by components, such as files and directories, can be easily copied from a gold server or data source. Others, such as IIS Web site settings or Windows registry entries, need to be intelligently extracted from a data source in order to be treated as an independent, manageable entity. With its built-in component types, the provisioning software can recognize the most common source items used for J2EE and Windows applications, and can intelligently and accurately extract data for use as a component source, store the component source in a repository, and install the source items correctly in its intended destination.

When a component contains a procedure, such as testing to verify that a web server is alive, the procedure is seen ion the procedures section of the components details page.

The provisioning software comes populated with a large number of commonly used component types. See “Built-in Components Types” on page 113 for more information on built-in component types.

Checking In a Component Type

Checking in a resource means copying a specific piece of software from a specific location (such as a directory on a gold server) and entering it in the repository with a specific name, a version number, and a resource type.

A resource type identifies the format and in many cases the function of a resource. Examples of resource types are file, directory, IIS Web Site Settings, and COM+ Application. For a complete list, see “Built-in Components Types” on page 113.

The resource repository uses a hierarchical namespace. Within this namespace, individual resources are identified by their name, which must be unique, and their version number.

The CLI command `cdb.rsrc.ci` checks in a resource. When you check in a resource with `cdb.rsrc.ci`, you specify the location of the source of the resource (with the `-src` argument), the location (a hierarchical name) in the repository where you would like to store the resource (with the `-dst` argument), and the resource type (with the `-type` argument).

A resource may be used for more than one component. Checking in a resource with `cdb.rsrc.ci` does not associate the resource with any particular component.

Extended Control Services

Extended control services are procedures that perform a software operation related to a resource or component. For a description of general purpose control services, see “General Purpose Extended Control Services” on page 153.

Built-in Components Types

Built-in component types enable you to quickly model many of the most common WebLogic, Windows, and J2EE application components and to automatically associate install, uninstall, export, and snapshot behavior with a particular resource.

TABLE 5–6 List of Built-in Component Types

Any WebLogic	Registry Key
WebLogic Enterprise Application	Windows Registry File
WebLogic Web Application	Data Source Name
WebLogic EJB	Windows Installer File (.msi)
Any IIS	Windows Batch file
IIS Application	Windows scripting host script
IIS Web Site or Virtual Directory Settings	Any UNIX
IIS Global Settings	Symbolic Link
Global ISAPI Filter Settings	RPM File
Web site ISAPI Filter Settings	File
Any Windows	Directory
COM+ Application	Container
COM Object	untyped

Component Type: File

Files represent an `untyped` single file taken from a target machine. The provisioning software deploys the files directly with no special post processing. Although files have a system component the functionality for installation, snapshot, and un-installation are built-in to the system and are not represented in the component XML.

Browsing

TABLE 5–7 File Browsing Behavior

	Unix	Windows
Root Path	"/"	List of physical drives on the host or network shares mapped to a drive letter. Removable media (floppy, cd, zip, etc.) are not displayed.
Delimiter	"/"	"\"
Ordering	Alphabetical with directories first	
Selection Type	User can single select a file for check in. Double click on a directory to view its contents	

TABLE 5–7 File Browsing Behavior *(Continued)*

Sample Path	/foo/foo.txt	C:\foo\foo.txt
Filters	None	
Special	Links will display their local name and the location pointed to. i.e. “foo->/usr/bar”	

Extended Control Procedures

TABLE 5–8 File Extended Control Procedures

Name	Parameters	Description
Delete	path: Full path to a file on disk	Removes the file from the filesystem. Provides a platform independent way to remove files.

Component Type: Directory

Directories represent an untyped collection of files and folders taken from a target machine. Although Directories have a system component the functionality for installation, snapshot, and un-installation are built-in to the system and are not represented in the component XML.

Browsing

TABLE 5–9 Directory Browsing Behavior

	Unix	Windows
Root Path	“/”	List of physical drives on the host or network shares mapped to a drive letter. Removable media (floppy, cd, zip, etc.) are not displayed.
Delimiter	“/”	“\”
Ordering	Alphabetical with directories first	
Selection Type	User can single select a directory for check in. Double click on a directory to view its contents	

TABLE 5–9 Directory Browsing Behavior *(Continued)*

Sample Path	/foo/foo	C:\foo\foo
Filters	None	
Special	Links will display their local name and the location pointed to. i.e. “foo/->/usr/bar/”	

Extended Control Procedures

TABLE 5–10 Directory Extended Control Procedures

Name	Parameters	Description
Delete	path: Full path to a directory on disk	Removes the directory from the filesystem and all if its children recursively. Provides a platform independent way to remove directories.

IIS Types Introduction

The following four IIS Types share a common implementation. All four export, Install, and delete data stored in the IIS metabase. As a result there are a common set of functions, formats, and errors.

Exported/Internal File Format

All four IIS types use an XML format to store their section of the metabase. The present XML format does not support metabase properties of type NTACL (such as AdminACL); if any are encountered while reading/writing to the metabase, they are ignored. Also, properties of type IPSec (such as IPSecurity) are written out as serialized objects, and as a result they are not human-readable (during either direct examination, or as difference results).

M-I Difference

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

Error Conditions

TABLE 5–11 IIS Types Introduction Error Conditions

Action	Condition	Result
Install/Export	IIS Does not exist or is the improper version	Install/Export fails
Install/Uninstall	Remote Agent does not have administrator privileges	Install/Uninstall fails

IIS Web Site or Virtual Directory Settings

Represents the settings for an IIS Website or Virtual Directory. Please note that this resource type only contains the settings for a website or virtual directory. The content on the web site must be checked in as a separate resource.

Browsing

TABLE 5–12 IIS Types Introduction Browsing Behavior

Root Path	List of web sites on the target system
Delimiter	"\"
Ordering	web sites and virtual directories appear in the order they occur in the metabase. This corresponds to the order they appear in the IIS Control panel and are NOT alphabetical.
Selection Type	User can single select a web site or virtual directory. Selecting a web site is considered recursive. Double-clicking on a web site provides a listing of the virtual directories underneath the web site.
Sample Path	Website1\VirturalDir2
Filters	None

Installation

Installation occurs by reading the XML file and importing into the target system metabase. If a web site with the same name exists it is overwritten. If there are multiple web sites with the same name on the system the first matching one will be removed and overwritten.

Special cases include untyped keys/nodes (see below for more info) and SSL certificates, which are not deployed. The relevant settings for SSL certs in IIS (SSLCertHash and SSLStoreName) are preserved during a deployment if they exist on the target, but if they do not exist they are not added.

To bring up a secure site after being deployed (or redeployed) requires a restart of IIS.

Un-Install

The entire web site is removed on the target system. All Virtual Directories underneath the web site are removed regardless of whether they were installed by the provisioning software. If the settings are just for a virtual directory only that directory is removed, not its containing site. Note the matching for uninstall is done via name, so the first web site with the same name found on the system will be uninstalled. Once this is complete the XML file used during installation is removed.

Component Type: IIS Global Filter Settings

Resource type is used to represent Global ISAPI filter settings. Please note that this resource type only contains the settings for a IIS Global Filter. The actual DLL implementing the filter must be installed separately.

Browsing

TABLE 5–13 IIS Global Filter Settings Browser Behavior

Root Path	Flat list of the global filters on the target system
Delimiter	N\A
Ordering	Filters appear in the order they occur in the metabase. This corresponds to the order they appear in the IIS Control panel. This is not alphabetical.
Selection Type	User can single select a filter. Filters cannot be expanded
Sample Path	Filter1
Filters	None

Installation

Installation occurs by reading the XML file and importing into the target system metabase. If a filter setting with the same name exists on the target machine it is overwritten.

Un-Install

The filter settings are removed on the target system then the XML file used during installation is removed.

Component Type: IIS Web Site Filter Settings

Please note that this resource type only contains the settings for a Web site Filter. The actual DLL implementing the filter must be installed separately.

Browsing

TABLE 5–14 IIS Web Site Filter Settings Browser Behavior

Root Path	List of the web sites on the system
Delimiter	\
Ordering	Filters will appear in the order they occur in the metabase. This will correspond to the order they appear in the IIS Control panel. This is not alphabetical.
Selection Type	User must first expand a web site to see the list of filters for that web site. All web site filters, or just an individual filter may be selected for check in.
Sample Path	Website1\filter1
Filters	None

Installation

Installation occurs by reading the XML file and importing into the target system metabase. If a filter settings with the same name exists on the target machine it is overwritten.

Un-Install

The filter settings are removed on the target system metabase, then the XML file used during installation is removed.

Component Type: IIS Global Settings

Resource type is used to represent Global IIS Settings.

Browsing

TABLE 5–15 IIS Global Settings Browser Behavior

Root Path	List of the global settings on the target system
Delimiter	N\A
Ordering	Settings will appear in the order they are presented in the metabase.
Selection Type	User can single select an individual setting for check in. Settings have no children and cannot be expanded.
Sample Path	AspCacheSize
Filters	none

Installation

Installation occurs by reading the XML file and importing into the target system metabase. The setting on the target system is overwritten if it exists.

Un-Install

Global settings cannot be uninstalled. Uninstalls will have no affect on the target system except to remove the *.XML file used during install.

Component Type: COM+ Application

Resource type is used to represent COM+ Applications. COM+ Applications are treated as a monolithic unit. The settings and content are installed as a group.

Browsing

TABLE 5–16 COM+ Application Browser Behavior

Root Path	List of COM+ Applications on the system
Delimiter	N\A
Ordering	Alphabetical based on application name.

TABLE 5–16 COM+ Application Browser Behavior *(Continued)*

Selection Type	User can single select an individual COM+ Application for check in. COM+ Applications have no children and cannot be expanded.
Sample Path	FM Stocks
Filters	None

Exported/Internal File Format

COM+ Applications are exported into a Windows Installer File (*.MSI) using the COM+ Admin SDK.

M-I Difference

The COM+ Application is re-exported on the target system as an MSI file and compared against the MSI file used to install the application. M-I diff will only indicate that there were differences (i.e. the two binary files are different) but will not indicate the details of the differences.

Installation

- If the COM+ Application with the same name is already installed on the target system and running as a service then it is stopped along with all of its running dependent services. The COM+ Application will then be deleted from the COM+ Catalog.
- The new COM+ Application is installed using the COM+ Admin SDK.
- To start the COM+ Application the user will have to use the startApplication call step to manually start the COM+ application.

Un-Install

The MSI used to install the COM+ Application is used to uninstall the COM+ Applications using the following command line:

```
msiexec /qn /x <path to msi file>
```

Once this is complete the msi file is removed from the target system.

Error Conditions

TABLE 5–17 COM+ Application Error Conditions

Action	Condition	Result
Install	COM+ Application already exists with the same name and either cannot be stopped or dependent services cannot be stopped.	Installation fails.
Uninstall	MSI file used for installation is no longer available	Uninstall fails
Install/Uninstall	Remote Agent does not have administrator privileges	Install/Uninstall fails

Extended Control Procedures

TABLE 5–18 COM+ Application Extended Control Procedures

Name	Parameters	Description
startApp	appName: Full name of the COM+ application.	Starts the COM+ application if it is run as a service
stopApp	appName: Fill name of the COM+ application to stop.	Stops the COM+ application and all dependent services
stopRouter	N/A	Stops the COM+ Routing services
startRouter	N/A	Starts the COM+ Routing services
installAsUser	rsrcSrcPath: Name of the COM+ application rsrcInstallPath: Path to the msi file representing the application userID: User whom is going to run the application password: Password of the user	Allows installation of a COM+ application which runs as a particular user.

Component Type: COM Component

Resource type is used to represent COM components.

Browsing

TABLE 5–19 COM Component Browser Behavior

Root Path	Uses standard File Browser
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.ocx. *.dll

Exported/Internal File Format

COM components are stored as a file in their native format.

M-I Difference

The COM component is compared as a binary file against the file used during installation. M-I diff will only indicate that there were differences (i.e. the two binary files are different) but will not indicate the details of the differences.

Installation

Regsvr32 is called to register the COM components in the dll using the following command line call:

```
regsvr32.exe /s <file path>
```

Uninstall

Regsvr32 is called to unregister the COM components in the dll using the following command line call:

```
regsvr32.exe /s /u <file path>
```

After the dll is unregistered it is removed from the target system.

Error Conditions

TABLE 5–20 COM Component Error Conditions

Action	Condition	Result
Install	The supplied .dll or .ocx does not contain COM components	Installation fails.
Uninstall	The supplied .dll or .ocx does not contain COM components	Uninstall fails

Component Type: Registry Keys

This resource type is used to represent registry keys and their associated values.

Browsing

TABLE 5–21 Registry Keys Browser Behavior

Root Path	List of the 5 main registry roots: HKEY_LOCAL_MACHINE HKEY_CLASSES_ROOT HKEY_CURRENT_USER HKEY_USERS HKEY_CURRENT_CONFIG
Delimiter	\
Ordering	Settings will appear in the order in which they are presented in the registry.
Selection Type	User can single select an individual key for check in. Selecting a key will check in that key and all of its children. Double clicking on keys will recursively check down the registry until a value is found. The name the value is displayed but not its contents. Values can be individually exported.
Sample Path	HKEY_LOCAL_MACHINE\Software\Example\Key
Filters	None

Exported/Internal File Format

Registry keys are exported into an XML file.

M-I Difference

During a snapshot the current state of registry key (and its children) is exported into an XML file. During an M-I difference the registry key is re-exported and compared against the original XML file. The standard XML diff comparator is used to generate differences between these files.

Installation

The XML file representing the registry is read and imported into the target system using an execJava step. Any keys are values already existing in the target system is overwritten.

Un-Install

The execJava implementation will take the root of the exported key, and delete all keys and values beneath it. If the root is a value, it will be deleted.

Error Conditions

TABLE 5–22 Registry Keys Error Conditions

Action	Condition	Result
Install/Uninstall	Remote Agent does not have administrator privileges	Install/Uninstall fails

Component Type: Registry File

Registry files are generated by Regedit to represent exports to the metabase. *.reg files are in text format and specify the keys and values to add or remove from the registry.

Browsing

TABLE 5–23 Registry File Browser Behavior

Root Path	Uses standard File Browser
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.reg

Exported/Internal File Format

COM components are stored as a file in their native text file format.

M-I Difference

M-I Differencing is not supported for *.reg files. Snapshots will not be taking during installation resulting in nothing to diff during the M-I diff. If the user would like to difference registry changes they are encouraged to use the built-in Registry keys type.

Installation

Regedit /s <file path> is called on the *.reg file to write its changes to the registry.

Un-Install

During uninstall only the *.reg file used during installation will be removed. The registry keys inside the reg file are unaffected. Users are encouraged to use the build-in Registry keys type to allow for registry un-installation.

Error Conditions

TABLE 5–24 Registry File Error Conditions

Action	Condition	Result

TABLE 5–24 Registry File Error Conditions *(Continued)*

Install	The supplied *.reg file is not in the proper format for regedit.	Installation fails.
Install	The agent does not have proper permissions to write into the registry sections designated by the *.reg file	Installation fails.

Component Type: Data Source Name

DSN Entries represent ODBC settings for connecting to a database. They can be edited by the user on the system by bringing up the "Data Source Administrator" control panel. The actual settings are stored in specific places in the registry. As a result, the Data source name Resource Type is built on top of the Registry keys resource type. The DSN Install, export, and uninstall directly use the facilities provided by the Registry Key resource handler. The DSN browser wraps the Registry browser to provide an experience closer to the "Data Source Administrator" control panel.

Browsing

TABLE 5–25 Data Source Name Browser Behavior

Root Path	List of the 2 DSN roots: User System
Delimiter	/
Ordering	Settings will appear in alphabetical order.
Selection Type	Double clicking on the System and User roots will provide a list of the DSN entries underneath. Users can then select a single entry for check in.
Sample Path	User/Oracle8
Filters	None

Exported/Internal File Format

On export the browser will export the key containing all the DSN settings, as well as the value of the same name in the 'ODBC Data Sources' key at the same level in the registry hierarchy.

Un-Install

The DSN uninstall is based on the registry uninstall with the caveat that the path being deleted is the key containing the DSN settings, but not the key the DSN GUI uses to display the available DSN settings. Special logic exists to delete this key as well. The semantics of this differ slightly from the registry uninstall semantics, though they use the same executor.

Error Conditions

The DSN system component directly calls the Install method of the registry system component. Please reference the Registry Key section of this document for further information on implementation and possible errors.

Component Type: Windows Installer File (*.msi)

Resource type used to represent Silent MSI files.

Browsing

TABLE 5–26 Windows Installer File Browser Behavior

Root Path	Uses standard File Browser
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.msi

Exported/Internal File Format

MSI files are stored as a file in their native format.

M-I Difference

M-I Differencing is not supported for Windows Installer files. Snapshots will not be taking during installation resulting in nothing to diff during the M-I diff. Since ROX does not have first-hand knowledge of the actions taken during the installer run it is not feasible to determine what needs to be captured.

Installation

The windows installer service is called on the msi file to import it into the target system with the following command:

```
misexec /qn /i <file path>
```

Un-Install

The windows installer service uninstalls is called on the msi file used during installation to uninstall the package using the following command:

```
msiexec /qn /x <file path>
```

After msiexec is called the msi file is removed.

Error Conditions

TABLE 5–27 Windows Installer File Error Conditions

Action	Condition	Result
Install	The supplied *.msi is not a proper windows installer file.	Installation fails.
Install	The agent does not have proper permissions to run installations	Installation fails.
Uninstall	The package has already been uninstalled	Uninstall fails

Component Type: Windows Batch File

Resource type used to represent *.bat and *.cmd files.

Browsing

TABLE 5–28 Windows Batch File Browser Behavior

Root Path	Uses standard File Browser
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.cmd, *.bat

Exported/Internal File Format

Windows batch files are stored as a file in their native text format.

M-I Difference

M-I Differencing is not supported for Windows Batch files. Snapshots are not taken during installation resulting in nothing to diff during the M-I diff.

Installation

The batch file is run during installation.

Un-Install

During uninstall the *.bat file is removed from the target system.

Error Conditions

TABLE 5–29 Windows Batch File Error Conditions

Action	Condition	Result
Install	The supplied batch file is not a valid batch file or contains errors.	Installation fails.

Component Type: Windows Scripting Host Script

Windows scripting host (WSH) scripts are text files either vbscript (*.vbs) or jscript (*.js) or contained within an XML project file (*.wsf).

Browsing

TABLE 5–30 Windows Scripting Host Script Browsing Behavior

Root Path	Uses standard File Browser
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.js, *.vbs, *.wsf

Exported/Internal File Format

Stored as a file in their native text format.

M-I Difference

M-I Differencing is not supported for WSH files. Snapshots will not be taking during installation resulting in nothing to diff during the M-I diff.

Installation

The WSH script is run via cscript.exe as follows:

```
cscript <file path>
```

Un-Install

During uninstall the script file is removed from the target host.

Error Conditions

TABLE 5–31 Windows Scripting Host Script Error Conditions

Action	Condition	Result
Install	The supplied file is not a valid wsf file or contains errors.	Installation fails.

Component Type: IIS Website (IIS Virtual Directory)

Browsing

Uses IIS Website Settings Browser.

Exported/Internal File Format

Composite Component: IIS Website/VDir Settings, IIS Virtual Directory Set, IIS Website Filter Set, Directory

Upon creation the contained entities (listed above) are created and linked to separately as appropriate.

Component Type: IIS Website Filter

Browsing

Uses IIS Website Filter Settings Browser.

Exported/Internal File Format

Composite Component: IIS Website Filter Settings, COM Object

Upon creation the contained entities (listed above) are created and linked to separately as appropriate.

Component Type: Symbolic Link

Browsing

Uses standard File Browser (with restricted export rules)

Exported/Internal File Format

Symlinks are an exception among simple component types in that they do not contain a resource. The data of a symlink is stored as a set of variables (one each for name and location) in the component.

Component Type: WebLogic WAR file

Browsing

Not directly browsable. Must be obtained through WebLogic web application browsers.

Exported/Internal File Format

This component has a dual nature. It can be either a .war archive, or the exploded version of that archive. Thus the file format is either an archive file in native format, or a package, respectively.

M-I Difference

Will use the standard file/directory MI diff approach.

Installation

Can not be directly installed. Must be installed as part of a WebLogic web application container. The file/directory is copied to the filesystem based on the install path, and then registered with the WL admin server.

Un-Install

Can not be directly uninstalled. The containing WebLogic web application container must be uninstalled, which will result in the removal of this file/dir.

Error Conditions

See “Error Conditions” on page 136.

Component Type: WebLogic web application settings

Browsing

Not directly browsable. Must be obtained through WebLogic web application browsers.

Exported/Internal File Format

On export the relevant settings for this webapp will be read from the admin server and stored as a custom config gen'd file.

M-I Difference

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

Installation

Can not be directly installed. Must be installed as part of a WebLogic web application container. Installation involves reading the post-configured file and applying all the settings to the admin server.

Un-Install

Can not be directly uninstalled. The containing WebLogic web application container must be uninstalled, which will result in the removal of this file and related settings of the webapp within WebLogic. Settings not specific to the webapp will not be removed.

Error Conditions

See Table 5–34.

Component Type: WebLogic Web Application

Browsing

Two browsers are supported. An admin server browser from which you can select one of the installed applications and its relevant settings, and a filesystem browser from which you can select the WAR file and a component without settings is created.

Admin Server Browser:

TABLE 5–32 WebLogic Web Application Browsing Behavior (Admin Server)

Root Path	List of Web Applications on the admin server
Delimiter	N/A
Ordering	Alphabetical based on app name
Selection Type	Single individual web app. Only standalone web applications will show up in this list. Any webapps that are part of an EAR file will not be displayed in, or selection allowed from, this list.
Sample Path	JChart
Filters	None

Filesystem Browser:

TABLE 5–33 WebLogic Web Application Browsing Behavior (File System)

Root Path	Uses standard File Browser (directories are valid selections)
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.war

Exported/Internal File Format

Composite Component: WebLogic web application container

Installation

Must be targeted at a WL target (server/cluster) which will install the component on that target, and install the contained reg component on the admin server of the target.

Un-Install

Untargets the webapp from the target, and if not currently targeted elsewhere, removes the registration comp from the admin server.

Error Conditions

TABLE 5–34 WebLogic Web Application Error Conditions

Action	Condition	Result
Install	The topology is incorrectly configured (target host doesn't point at correct domain host)	Targeting fails.
Install	The target host is not a valid WL target.	Installation prohibited.
Browsing/Install/Uninstall	Credentials aren't properly configured.	Operation fails.
Browsing	Path not correctly configured in domain host.	Browsing fails.

Component Type: WebLogic JAR file

Browsing

Not directly browsable. Must be obtained through WebLogic EJB browsers.

Exported/Internal File Format

This component has a dual nature. It can be either a.jar archive, or the exploded version of that archive. Thus the file format is either an archive file in native format, or a package, respectively.

M-I Difference

Will use the standard file/directory MI diff approach.

Installation

Can not be directly installed. Must be installed as part of a WebLogic EJB container. The file/directory is copied to the filesystem based on the install path, and then registered with the WL admin server.

Un-Install

Can not be directly uninstalled. The containing WebLogic EJB container must be uninstalled, which will result in the removal of this file/dir.

Error Conditions

See “Component Type: WebLogic EJB” on page 138.

Component Type: WebLogic EJB settings

Browsing

Not directly browsable. Must be obtained through WebLogic EJB browsers.

Exported/Internal File Format

On export the relevant settings for this EJB will be read from the admin server and stored as a custom config gen’d file.

M-I Difference

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

Installation

Can not be directly installed. Must be installed as part of a WebLogic web application container.Installation involves reading the post-configured file and applying all the settings to the admin server.

Un-Install

Can not be directly uninstalled. The containing WebLogic web application container must be uninstalled, which will result in the removal of this file and related settings of the webapp within WebLogic. Settings not specific to the webapp will not be removed.

Error Conditions

See “Component Type: WebLogic EJB” on page 138

Component Type: WebLogic EJB container

Browsing

Not directly browsable. Must be obtained through WebLogic EJB browsers.

Exported/Internal File Format

Composite Component: WebLogic JAR file, WebLogic EJB container

Installation

Must be targeted at a WL Domain host. Installs the nested components, and registers the EJB with WebLogic. Can be installed as part of a retarget during installation of a WebLogic EJB.

Un-Install

Removes the EJB from WebLogic, then uninstalls the nested components.

Error Conditions

TABLE 5–35 WebLogic EJB container Error Conditions

Action	Condition	Result
Uninstall	A dependant WebLogic EJB is still installed.	Uninstall fails indicating the dependency.

Component Type: WebLogic EJB

Browsing

Two browsers are supported. An admin server browser from which you can select one of the installed applications and its relevant settings, and a filesystem browser from which you can select the JAR file and a component without settings is created.

Admin Server Browser:

TABLE 5–36 WebLogic EJB Browsing Behavior (Admin Server)

Root Path	List of EJBs on the admin server
Delimiter	N/A
Ordering	Alphabetical based on app name

TABLE 5–36 WebLogic EJB Browsing Behavior (Admin Server) *(Continued)*

Selection Type	Single individual web app. Only standalone EJBs will show up in this list. Any EJBs that are part of an EAR file will not be displayed in, or selection allowed from, this list.
Sample Path	companyStoreEJBs
Filters	None

Filesystem Browser:

TABLE 5–37 WebLogic EJB Browsing Behavior (File System)

Root Path	Uses standard File Browser (directories are valid selections)
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.jar

Composite Component: WebLogic EJB Container

Installation

Must be targeted at a WL target (server/cluster) which will install the component on that target, and install the contained reg component on the admin server of the target.

Un-Install

Untargets the EJB from the target, and if not currently targeted elsewhere, removes the registration comp from the admin server.

Error Conditions

TABLE 5–38 WebLogic EJB Container Error Conditions

Action	Condition	Result
Install	The topology is incorrectly configured (target host doesn't point at correct domain host)	Targeting fails.
Install	The target host is not a valid WL target.	Installation prohibited.
Browsing/Install/Uninstall	Credentials aren't properly configured.	Operation fails.
Browsing	Path not correctly configured in domain host.	Browsing fails.

Component Type: WebLogic EAR file

Browsing

Not directly browsable. Must be obtained through WebLogic enterprise application browsers.

Exported/Internal File Format

This component has a dual nature. It can be either a .EAR archive, or the exploded version of that archive. Thus the file format is either an archive file in native format, or a package, respectively.

M-I Difference

Will use the standard file/directory MI diff approach.

Installation

Can not be directly installed. Must be installed as part of a WebLogic enterprise application container. The file/directory is copied to the filesystem based on the install path, and then registered with the WL admin server.

Un-Install

Can not be directly uninstalled. The containing WebLogic EJB container must be uninstalled, which will result in the removal of this file/dir.

Error Conditions

See Table 5–34

Component Type: WebLogic enterprise application settings

Browsing

Not directly browsable. Must be obtained through WebLogic enterprise application browsers.

Exported/Internal File Format

On export the relevant settings for this app will be read from the admin server and stored as a custom config gen'd file.

M-I Difference

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

Installation

Can not be directly installed. Must be installed as part of a WebLogic enterprise application container. Installation involves reading the post-configured file and applying all the settings to the admin server.

Un-Install

Can not be directly uninstalled. The containing WebLogic enterprise application container must be uninstalled, which will result in the removal of this file and related settings of the webapp within WebLogic. Settings not specific to the webapp will not be removed.

Error Conditions

See Table 5–34

Component Type: WebLogic Enterprise Application Container

Browsing

Not directly browsable. Must be obtained through WebLogic enterprise application browsers.

Exported/Internal File Format

WebLogic EAR file, WebLogic enterprise application container, WebLogic List.

Installation

Must be targeted at a WL Domain host. Installs the nested components, and registers the Enterprise Application with WebLogic. Can be installed as part of a retarget during installation of a WebLogic enterprise application, or WebLogic Contained module.

Un-Install

Removes the Enterprise Application from WebLogic, then uninstalls the nested components.

Error Conditions

TABLE 5–39 WebLogic Enterprise Application Container Error Conditions

Action	Condition	Result
Uninstall	A dependant WebLogic enterprise application, or contained WebLogic module is still installed.	Uninstall fails indicating the dependency.

Component Type: WebLogic Enterprise Application

Browsing

Two browsers are supported. An admin server browser from which you can select one of the installed applications and its relevant settings, and a filesystem browser from which you can select the EAR file and a component without settings is created.

Admin Server Browser:

TABLE 5–40 WebLogic Enterprise Application Browsing Behavior (Admin Server)

Root Path	List of Enterprise Applications on the admin server
Delimiter	N/A
Ordering	Alphabetical based on app name
Selection Type	Single individual enterprise application.
Sample Path	companyStoreAdmin
Filters	None

Filesystem Browser:

TABLE 5–41 WebLogic Enterprise Application Browsing Behavior (File System)

Root Path	Uses standard File Browser (directories are valid selections)
Delimiter	
Ordering	
Selection Type	
Sample Path	
Filters	*.ear

Exported/Internal File Format

Composite Component: WebLogic enterprise application container.

Installation

Must be targeted at a WL target (server/cluster) which will install the component on that target, and install the contained reg component on the admin server of the target.

Un-Install

Untargets the enterprise app from the target, and if not targeted elsewhere, removes the registration comp from the admin server.

Error Conditions

TABLE 5–42 WebLogic Enterprise Application Error Conditions

Action	Condition	Result
Install	The topology is incorrectly configured (target host doesn't point at correct domain host)	Targeting fails.
Install	The target host is not a valid WL target.	Installation prohibited.
Browsing/Install/Uninstall	Credentials aren't properly configured.	Operation fails.
Browsing	Path not correctly configured in domain host.	Browsing fails.

Component Type: WebLogic List

Browsing

Not directly browsable. Must be obtained through WebLogic enterprise application browsers.

Exported/Internal File Format

This component is a container component with some custom install/uninstall/snapshot blocks. In all other ways it is only a generic container component.

Installation

Does not support direct installation.

Un-Install

Does not support direct uninstallation.

Error Conditions

See contained WebLogic module types.

Component Type: Contained WebLogic web Application Container

Browsing

Not directly browsable. Must be obtained through WebLogic enterprise application browsers.

Exported/Internal File Format

Composite Component:

Contains a final WebLogic WAR file stub component, and a non-final, stub by default, WebLogic web application settings component.

Also contains a final compRef to the WebLogic enterprise application container component contained by the WebLogic enterprise application that contains the WebLogic List that contains this component.

Installation

Can be installed on a WL Domain host. Results in the installation of the WebLogic enterprise application container as well.

Un-Install

Uninstalls the WebLogic enterprise application container.

Error Conditions

TABLE 5–43 Contained WebLogic web Application Container Error Conditions

Action	Condition	Result
Uninstall	A dependant contained WebLogic web application is installed.	Uninstall fails indicating the dependency.

Component Type: contained WebLogic web application

Same as WebLogic web application except it contains a final Contained WebLogic web application container.

Component Type: Contained WebLogic EJB Container

Browsing

Not directly browsable. Must be obtained through WebLogic enterprise application browsers.

Exported/Internal File Format

Composite Component:

Contains a final WebLogic JAR file stub component, and a non-final, stub by default, WebLogic web application settings component.

Also contains a final compRef to the WebLogic enterprise application container component contained by the WebLogic enterprise application that contains the WebLogic List that contains this component.

Installation

Can be installed on a WL Domain host. Results in the installation of the WebLogic enterprise application container as well.

Un-Install

Uninstalls the WebLogic enterprise application container.

Error Conditions

TABLE 5–44 Contained WebLogic EJB Container Error Conditions

Action	Condition	Result
Uninstall	A dependant contained WebLogic web application.	Uninstall fails indicating the dependency.

Component Type: Contained WebLogic EJB

Same as WebLogic EJB except it contains a final Contained WebLogic web application container.

System Components

In addition to the Component Types described above System Components also provide functionality available to advanced plan and component authors. The following components have additional functionality:

Windows System Service

The Windows System Component provides utilities for interacting with different windows systems, beyond those related to any specific component type described above.

Control Procedures

TABLE 5–45 Contained WebLogic EJB Control Procedures (Windows)

Name	Parameters	Description
stopIIS	None	Shuts down the IIS process.
startIIS	None	Restarts IIS if it was not running
stopService	serviceName: Name of the service to stop.	Shuts down the service.

TABLE 5–45 Contained WebLogic EJB Control Procedures (Windows) *(Continued)*

startService	serviceName: Name of the service to start.	Starts the windows service

WebLogic Target

The WebLogic Target component provides utilities for interacting with WebLogic managed servers and clusters.

Procedures

TABLE 5–46 Contained WebLogic EJB Control Procedures (WebLogic)

Name	Parameters	Description
Start	None	Starts the target.
Stop	None	Stops the target.
Restart	Boolean restart	Restarts the target, based on the value of the restart flag.

Component Type User Interface

The component types user interface provides the functionality for users to manage component types. Users can check-in component types, view lists of all component types, view component type versions. Users can also view install information for component types such as which hosts a component is installed on and which plan installed the component.

The HTML User Interface also provides browsing of several component types. This allows for the selection of component types for remote check in.

The Component Type Page

The component type page allows you to list component types, view component type details, and create component types. This page also contains controls for filtering the component type list and selecting sort criteria. You can change the sort criteria for the component type list by clicking on one of the sort arrows in the header area. Click either the ascending or descending arrow selects that column for sorting and the sort order.

Component Types	A user definable text string that names the component object. By default, the components are listed in alphabetical order by component name. You can sort the component type list by its name by clicking on one of the sort arrows in the header area for this column. Click either the ascending or descending arrow selects this column for sorting and the sort order.
Menu Group	The component type menu is divided into several sections or groups. This field identifies which section or group within the pull down menu the component type belongs. Note: Other than the *hidden* menu group, this field is only used as a sort criteria and does not effect the menu list. Assigning a component type to the hidden menu group overrides the menu order causing the component type not to appear in the menu.
Menu Order	This field controls where in the component type menu a component type name appears. The provisioning software simply sorts the menu list in alpha numerical order based on this field. You can use both number and letters in this field. However, the provisioning software has only used numbers by convention. To help organize the data the provisioning software has also adopted the following numbering convention. Assigning a component type a menu group of *Any WebLogic* does not effect where in the menu the component type will appear. However, by following the arbitrary convention set by the provisioning software, assigning a component type that belongs with the menu group of *Any WebLogic* the 000100-xxxxxx-xxxxxx will put it in the menu in the correct place.
Indent Level	This defines how many units of 2 spaces to the right the menu item will be indented.
Description	An optional user defined text string that describes the component object. This attribute is not used by the provisioning software application but can provide meaningful information to the user.
Defining Component	Displays the component that defines this component type.
Version	Displays the revision number of the component. Each time a component is modified, its version number is incremented.

	You can sort the component type list by version number by clicking on one of the sort arrows in the header area for the column. Click either the ascending or descending arrow selects this column for sorting and the sort order.
Action	Links that take you to other component type related pages to either create a new component type, view more information about a specific component type, or edit component types.

The Component Type Details Page

The component types details page allows you to view Component types information and attributes. None of the fields can be changed on this page.

Component Type	Displays the name of the component type.
Menu Group	Displays the name of the menu group.
Menu Order	Displays 18 character code that controls where in the menu this component type is placed.
Indent Level	Displays how many spaces this menu item is indented.
Description	Displays the description of the component type.
Defining Component	Displays the component that defines this component type.
Version	Displays the version number for this component type.
Edit Button	A link that takes you to the component types edit page.
Done Button	A link that takes you to the component types page.

The New Component Type Edit Page

"Working with Components" on page 91 shows the Component Edit page when creating a component. This edit page is different from the on used to edit existing components.

Component Type	Displays the name of the component type. This cannot be changed.
Menu Group	A text field that allows you to change the name of the menu group for this component type. Only predefined menu groups are allowed.
Menu Order	A text field that allows you to change the location in the menu for this component type. This field accepts any alpha or numeric character.

Indent Level	A text field that allows you to change the indentation on the menu for this component type. This field accepts number in the range of 0 through 10.
Description	A text field that allows you to change the description for this component type.
Defining Component	Displays the component that defines this component type.The View Component is a link that take you to the component details page for the defining component.
Save Button	Saves the contents of the edit page incrementing the version number.
Cancel Button	A link that takes you to the component types page.

▼ How To View Component Types

Steps

1. **In the left-hand navigation menu, click administrative.**
   The HTML user interface displays the administrative page, which allows the user to select one of four administrative functions.
2. **Click on the component types link on the administrative page.**
   The HTML user interface displays a list of the component types.

▼ How To View Component Types Details

Steps

1. **In the left-hand navigation menu, click administrative.**
   The HTML user interface displays the administrative page, which allows the user to select one of four administrative functions.
2. **Click on the component types link on the administrative page.**
   The HTML user interface displays a list of the component types.
3. **Find the row listing the component type you're interested in, and click details.**
   The HTML user interface displays a Components Types Details page for the component type.

▼ How To Create Component Types

Steps

1. **In the left-hand navigation menu, click administrative.**

   The HTML user interface displays the administrative page, which allows the user to select one of four administrative functions.

2. **Click on the component types link on the administrative page.**

   The HTML user interface displays a list of the component types.

3. **In the top row of the table listing components types, enter a name in the component types field for the new component type, and click create.**

   The HTML user interface displays the Components Types Details Edit page for the new component type.

4. **Use the controls on this page to specify the necessary information that will make up the component type.**

   a. **Change the name in the component field, if desired. Once the component is checked in you will not be able to change the name.**

   b. **Enter a Menu Group name.**

   c. **Define the menu order by entering a 18 digit number.**

   d. **Specify the indent level.**

   e. **Enter a description in the Description field.**

   f. **Enter a description if one is desired.**

   g. **Select a defining component.**

   h. **When finished click** *save***.**

▼ How To Edit Component Types

Steps

1. **In the left-hand navigation menu, click administrative.**

   The HTML user interface displays the administrative page, which allows the user to select one of four administrative functions.

2. **Click on the component types link on the administrative page.**

   The HTML user interface displays a list of the component types.

3. **Find the row listing the component type you're interested in, and click details.**

The HTML user interface displays a Components Types Details page for the component type.

4. **Click the edit button located near the bottom of the page.**

   The HTML user interface displays a Components Types Details Edit page for the component type.

5. **Use the controls on this page to change any information that you want to change.**

   - Enter a Menu Group name.
   - Define the menu order by entering a 18 digit number.
   - Specify the indent level.
   - Enter a description in the Description field.
   - Enter a description if one is desired.
   - Select a defining component.
   - When finished click *save*.

General Purpose Extended Control Services

Extended control services are procedures that perform a software operation related to a resource or component. These services include services for starting and stop the Microsoft IIS Web Server and for starting and stopping specific Windows services. The provisioning software includes two sets of extended control services that are not specific to individual resource types:

- general purpose control services, which can be called by any component or plan
- resource-type-specific control services, which are related to a specific type of resource and that can be called only by components or plans that include resources with those resource types

General purpose control services include the ability to start or stop Windows services. These control services can be called by any component or plan.

An example of a resource-type-specific control service is the `startRouter` procedure included with the COM+ Application resource type. This procedure starts COM+ routing services. It works only with resources that are of the COM+ Application resource type.

Extended Control Services for IIS

Table 5–47 lists the provisioning software's built-in procedures for controlling IIS.

TABLE 5–47 Built-in Procedures for Controlling IIS

Name	Parameters	Description
stopIIS	None	Shuts down the IIS process.
startIIS	None	Restarts IIS if it is not running

Extended Control Services for Windows Services

A component or plan may include steps to start or stop specific Windows Services.

Table 5–48 describes provisioning software's built-in procedures for controlling Windows Services.

TABLE 5–48 Built-in Procedures for Controlling Windows Services

Name	Parameters	Description
stopService	serviceName: Name of the service to stop.	Shuts down the service.
startService	serviceName: Name of the service to start.	Starts the windows service

CHAPTER 6

Plans

A plan is an XML document that performs operations on hosts specified by the operator.

This section discusses the following topics:

Working with Plans

A plan is a sequence of instructions typically used to manipulate one or more components. For example, an execution plan may instruct the provisioning software to install three components and initiate the "startup" control on another. A plan can also be defined as a sequence of other plans, which allows common instruction sequences to be shared between multiple plans.

The provisioning software provides an in-memory representation of the objects expressed by the plan and component language and also defines a process for the validation, persistence and version control for those objects. The provisioning software manages the physical resources associated with a component. When the provisioning software executes a plan, substitution variables declared by a component are replaced. The Notification feature can be used to send email in response to plan execution related events. The provisioning software also includes a number of predefined components that may be used directly, or as samples for constructing other components.

Plan Types

Within the N1 Service Provisioning System software mode there are two types of plans:

- Composite plans, which contain only subplans.
- Simple plans, which contain a collection of simple steps, but which cannot call other plans.

The XML Schema enforces these restrictions so that you can either have a top-level plan which only calls other subplans, or you can have a simple plan with various simple steps but no calls to other subplans. The distinction is important because the steps contained in a simple plan all execute on the same set of target hosts, whereas the steps of a composite plan may execute on different sets of target hosts, using one set of target hosts for each simple plan recursively contained in the composite plan.

Types of Steps

Steps are logical blocks that manage what to do with subplans or components. The N1 Service Provisioning System software supports three types of steps:

- Steps that can be called only from a component. These steps can in turn be divided into two types:
  - Component steps that can only be called from an install block.
  - Component steps that can be called from outside an install block.
- Steps that can be called only from plans. These steps consist of two groups:
  - Steps that can be called only from composite plans.
  - Steps that can be called only from simple plans.
- Steps that can be called from both components and plans

Plan XML Schema

For a detailed description of the XML schema for plans, see *N1 Service Provisioning System 4.1 Reference Guide*.

Plans User Interface

The Plans Page

The plans page allows you to list plans, view plan details, create plans, and edit plans. This page also contains controls for filtering the component list and selecting sort criteria. There are blue wedge shaped arrows over several of the columns on the plans page that allow you to select the column to sort the list by and in either ascending or

descending order. When one of the arrows is clicked it selects that column by which to sort, sorts the list in either ascending or descending order (depending on which arrow was clicked), and changes the arrow to white . The white arrow indicates which column was used for the sort and in which order (ascending or descending).

Path	Allows you to specify where to look for or store plans. Click on *change path* to specify a different path.
Show	Allows you to list either plans or components. If you select components, the HTML user interface displays the components page as though you had click on the components option in the left-hand navigation menu. You can also narrow the list by selecting a category from the category pull down menu.
(Check Box)	Marks a plan for inclusion when one of the actions are clicked in the *Actions for Checked plans* area. When clicked, *Actions for Checked plans* will either delete or categorize a plan.
Plan	A user definable text string that names the plan object. By default, the plans are listed in alphabetical order by plan name. You can change the sort order between ascending and descending by clicking on one of the sort icons that is in the plans header area.
Version	Displays the revision number of the plan. Each time a plan is modified, its version number is incremented. You can sort the plan list by version number by clicking on one of the sort icons in the header area for this column. Clicking this icon toggles the order between ascending and descending.
Description	An optional user defined text string that describes the plan object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other plan related pages to either find a specific plan, create a new plan, view more information about a plan, or view the run history for a plan.
Actions for Checked Plans	From this area you can either delete or apply categories to the checked plan(s).

The Plans Details Page

The Plans Details page allows you to view a Plan's information and attributes. This page includes information and attributes information, and also contains buttons that allow you to run, edit, and delete a plan.

Plan	The name of the plan.
Version	Displays the revision number of the plan.
Checked In	Displays the date and time when the plan was checked in. That is, created or modified.
Checked In By	Displays the user ID of the one who checked in the plan. This provides an audit trail when trying to troubleshoot problems or inconsistencies.
Category	A user definable object that is used to filter the plan list. Categories are created using the categories page accessed from the Administrative page. Categories are then applied to a plan from the plans page. Categories can also be created by clicking in the "*Apply Categories . . .*" link.
Description	An optional user defined text string that describes the plan object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Buttons	By clicking one of these buttons you initiate one of the following functions.

Run	Runs the plan.
Advanced Edit	Allows you to edit the XML code for the plan.
Delete	Removes the plan from the database. If there is more than one version then the provisioning software asks if you want to delete just this version or all versions.
Done	Takes you to the plans page.
Download	Allows you to save the XML code to a text file. Between this function and the *Replace Definition With File* function the provisioning software allows you to use other text editors to create and modify the XML code for plans.

The Plans Advanced Edit Page

The Plan Advanced Edit page allows you to change what a plan does by changing the XML code of a plan.

Plan	Allows you to change the name of the plan.
Version	Displays the revision number of the plan. This will increment when the plan is checked in. This cannot be changed by the user.
Checked In	This will reflect a new time and date when the plan is checked in. This cannot be changed by the user.
Checked In By	This will reflect the user ID of the one who checks in the plan providing an audit trail. This cannot be changed by the user.
Category	Displays the category for the plan. Categories can be assigned or changed at from the plans page by clicking in the "*Apply Categories . . .*" link.
Edit The Plan Definition	This is a text field that displays the XML code defining what the plan does. You can make any desired changes directly into this text field.
Replace Definition With File:	This is a text field that allows you to replace the XML code in the plan definition field with that of a text file. Either type in the name of the file, including the path, into the text field or use the browse button to locate a file. Initiate this feature by clicking on the *Replace* link just to the right of the browse button.
Buttons	By clicking one of these buttons you initiate one of the following functions.

Check in	Allows you to check in the plan using its current name.
Check in as	Allows you to check in the plan using a new name.
Cancel	Takes you to the plans page without saving any changes.

▼ How To View Plans List

Steps

- **In the left-hand navigation menu, click plans.**

  The HTML user interface displays the plans page, which lists the plans already checked in to the database. By clicking on one of the sort arrows in the header yo can change the sort criteria.

▼ How To View Plan Details

Steps

1. **In the left-hand navigation menu, click plans.**

   The HTML user interface displays the plans page, which lists the plans already checked in to the database.

2. **Find the row listing the plan you're interested in, and click details.**

   The HTML user interface displays a Plan Details page for the plan.

▼ How To Edit Plans

Steps

1. **In the left-hand navigation menu, click components.**

   The HTML user interface displays the plans page, which lists the plans already checked in to the database.

2. **Find the row listing the plan you're interested in, and click details.**

   The HTML user interface displays a Plan Details page for the plan.

3. **Click the** *advanced edit* **button.**

   The HTML user interface displays a Plan Advanced Edit page for the plan.

4. **In the text box, make the changes you want to the plan's XML code, or enter new XML for the plan that you have authored in another program.**

   To upload a plan stored on your local system, enter the path of the file or use the browse button, then click Replace.

5. **Click the check in button to use the same name or click the check in as button to use a different name.**

   The HTML user interface displays the plans page, which lists the plans already checked in to the database.

   The HTML user interface displays a new page, telling you that you are about to check in a plan and increments the version number.

6. **Click continue to check in to finish checking in the plan.**

How To Create Plans

You can create an auto-generated plan, create a completely custom plan, or start with an auto-generated plan and customize it.

The N1 Service Provisioning System software can automatically generate a plan consisting of multiple “direct run” procedures. You can run this plan directly or save it for use as a template for more sophisticated plans that you author in XML.

For deployments that involve the coordination of multiple components, multiple host sets, or both, write a plan and use the provisioning software XML schema to define operations such as dependency checks and scripting that executes commands on an application console. Once the plan is written, you need to check it into the plan repository.

▼ How To Create Auto Generated Plans

Steps

1. **In the left-hand navigation menu, click components.**

   The HTML user interface displays the components page, which lists the components already checked in to the database.

2. **In the table listing components, find the row describing the component you would like to deploy, and click details.**

   The HTML user interface displays a Components Details page for the component.

3. **In the component procedures table, click the checkbox next to the name of each procedure you want to include in the plan.**

4. **When you have selected all the procedures you want to include, click the link labeled generate plan with checked procedures.**

   The HTML user interface displays a Plan Advanced Edit page for the plan.

5. **Click check in.**

▼ How To Create Custom Plans

You can begin by writing the XML code for the plan using a schema-validating editor such as TurboXML or you can write the XML code within the HTML user interface. The following description assumes that you will write the XML code using an external editor.

Steps

1. **Using a schema-validating editor, write your plan using the XML schema elements described in** *N1 Service Provisioning System 4.1 Reference Guide***.**

2. **When you have finished writing the plan, log in through your web browser.**

3. **In the left-hand navigation menu, click plans located under the** *application deployment* **heading.**

The HTML user interface displays the plans page, which lists the plans already checked in to the database.

4. **In the top row of the table listing plans, enter a name and a brief description for the plan you want to create, and click create.**

   The HTML user interface displays a Plan Advanced Edit page for the plan . Note that the provisioning software has populated the plan definition with an XML skeleton.

   In the text box, enter the XML for the plan. To upload a plan stored on your local system, enter the path of the file or use the browse button, then click Replace.

5. **Click the check in button.**

   The HTML user interface displays a new page, telling you that you are about to check in a plan. It assigns the plan version 1.0.

6. **Click continue to check in to finish checking in the plan.**

Summary of CLI Commands

Table 6–1 lists the CLI commands you can use to manage plans.

TABLE 6–1 Summary of the pdb Commands

Command	Description
pdb.p.ci	Checks in a new version of a plan.
pdb.p.co	Checks out a plan (outputs a plan in XML).
pdb.p.genplan	Generates and outputs a plan in XML.
pdb.p.la	Lists the latest versions of all plans.
pdb.p.lo	Views a plan.
pdb.p.lv	Lists all the versions of the specified plan.
pdb.p.del	Deletes an execution plan.
pdb.p.sh	Shows or hides a plan.
pdb.p.sc	Associates a plan with a set of categories.

Note – To run plans, use the `pe` commands described in Table 6–2.

Session Variables

Session variables are global values associated with deployments that expands the flexibility of the provisioning software. Session variables can be created and saved for future use with any deployment or created and not saved for use with a single deployment.

Session Variables User Interface

The Session Variables Page

The Session Variables page allows you to list, create, edit and, view the details of Sessions Variables.

(Check Box)	Marks a session variable for inclusion when one of the actions are clicked in the *Actions for Checked Session Variables* area. When clicked, *Actions for Checked Session Variables* will either delete the session variable.
Session Variable	A user definable text string that names the session variable object.
Password	Indicates if a password is required to change the session variable.
Value	Displays the value of the session variable. If a password is required for the session variable the value is seen as a series of ellipses
Action	Links that take you to other session variables related pages to either create or edit a session variable.
Actions for Checked Plans	From this area you can delete the checked session variable(s).

The Session Variables Details Page

This page is used to view the details for a session variables.

Session Variable	Displays the name of the session variable object.
Description	An optional user defined text string that describes the session variable. This attribute is not used by the provisioning software but can provide meaningful information to the user.

Value	Displays the value of the session variable unless it is password protected.
Session Variables	Displays the name of the session variable object.
Buttons	By clicking one of these buttons you initiate one of the following functions.
	Delete — Allows you to permanently delete the session variable.
	Edit — Takes you to the edit page for the session variable.
	Done — Takes you to the session variables page without saving any changes.

The Session Variables Edit Page

This page is used to create and edit session variables. If you are editing an existing session variable the top two items are displayed and cannot be changed.

Session Variable	A text field that allows you to enter a name for the session variable. If this is an existing session variable then this field only displays the name. It cannot be changed.
(Check Box)	Marks this session variable as password protected. When checked, the value of the session variable will be seen as a series of ellipses.
Description	A text field that allows you to enter an optional text string that describes the session variable. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Value	A text field that allows you to enter a value of the session variable.
Password	A text field that allows you to enter a password for the session variable.
Buttons	By clicking one of these buttons you initiate one of the following functions.
	Save — Saves the changes and takes you to the session variables page.
	Cancel — Takes you to the session variables page without saving any changes.

▼ How To View Session Variables

Steps

- **In the left-hand navigation menu, click session variables.**

  The HTML user interface displays the session variables page, which lists the session variables already checked in to the database.

▼ How To View Session Variables Details

Steps

1. **In the left-hand navigation menu, click session variables.**

   The HTML user interface displays the session variables page, which lists the session variables already checked in to the database.

2. **Find the row listing the session variable you're interested in, and click details.**

   The HTML user interface displays a session variables Details page for the session variable.

3. **Near the bottom of the page, click the *done* button when finished.**

   The HTML user interface displays a session variables page.

▼ How To Edit Session Variables

Steps

1. **In the left-hand navigation menu, click session variables.**

   The HTML user interface displays the session variables page, which lists the session variables already checked in to the database.

2. **Find the row listing the session variable you're interested in, and click details.**

   The HTML user interface displays a session variables Details page for the session variable.

3. **Near the bottom of the page, click the *edit* button.**

   The HTML user interface displays the session variables edit page for the session variables.

4. **Using the text fields make any desired changes.**

5. **Near the bottom of the page, click the *save* button when finished.**

   The HTML user interface save the changes and displays a session variables page.

6. **Near the bottom of the page, click the *cancel* button to cancel any changes.**

The HTML user interface displays a session variables page without saving any changes.

Running Plans

Once you have checked in a plan, you can run it in order to perform operations on hosts.

▼ How To View Running Plans

Steps

1. **In the left-hand navigation menu, click plans.**

   The HTML user interface displays the plans page, which lists the plans already checked in to the database.

2. **Find the row listing the plan you're interested in, and click details.**

   The HTML user interface displays a Plan Details page for the plan.

3. **Near the bottom of the page, click the *run* button.**

   The HTML user interface displays a Plan Details Run page.

4. **Enter the required information and click *run preflight only* to verify that the plan will run successfully or click *run plan (includes preflight)* to run the plan.**

▼ How To Use a Component's Direct Run Procedures

For straightforward deployments that don't involve multiple components or synchronization among hosts, you may not need to write a plan at all. Instead, you may be able to use one of the "direct run" procedures that the provisioning software automatically generated when you checked in the component you want to deploy.

To use one of these "direct run" procedures, do the following:

Steps

1. **In the left-hand navigation menu, click components.**

   The HTML user interface displays the components page, which lists the components already checked in to the database.

2. **In the table listing components, find the row describing the component you would like to deploy, and click details.**

   The HTML user interface displays a Components Details page for the component.

3. **In the component procedures table, click the checkbox next to the name of the procedure you want to run, and then click run.**

   The HTML user interface displays the Plan Details Run page for the procedure.

4. **Use the fields on the page to enter the variables and select the parameters you want to use when you run the plan.**

   - In the plan parameters area, use the target host field to select the host on which you want to run this plan, or use the pull-down menu to select a target host set.
   - In the plan variables area, configure the variables you want to set. Whether there are variables to configure depends on the contents of the plan.
   - In the options area, select whether you want to perform a detailed preflight. Running a detailed preflight increases the amount of time preflight takes, but also increases the detail the preflight goes into.
   - In the limits area, limit the number of hosts running at the same time by entering the number of hosts in the field provided. You can also limit overall running time of plan or limit running time of native calls by entering a number in the fields provided, and then selecting the time unit, such as "minutes," from pull-down menus.

5. **Do one of the following:**

   - **To run just the preflight test of the procedure, click run preflight only.**
   - **To run the preflight test and then (assuming the preflight test finds no errors) to run the procedure itself, click run plan (includes preflight).**

Summary of CLI Commands

Table 6–2 describes the CLI commands associated with running plans.

TABLE 6–2 Summary of the pe Commands

Command	Description
pe.h.prep	Prepares a set of hosts.
pe.p.en	Displays the output of an execNative step.
pe.p.la	Lists running and completed plans.
pe.p.lo	Lists information about a running or completed plan.

TABLE 6–2 Summary of the pe Commands *(Continued)*

Command	Description
pe.p.del	Deletes the history of a completed plan run.
pe.p.lp	Lists the subplans and targets associated with a plan.
pe.p.run	Runs a plan.
pe.p.stop	Stops a plan that is running
pe.pi.lo	Lists the parameters used to run a plan.

CHAPTER 7

Comparisons

Comparisons are used to validate an installation or system configuration. By comparing the configuration of a host against its original configuration you can tell if anything has changed. You can also compare one or more hosts against a reference host to ensure that all hosts are configured the same. With the N1 Service Provisioning System software you can compare one or more hosts against another host, one or more hosts against a model in the Master Server, or a host against its original configuration. The provisioning software allows you to check the entire configuration of a host or limit what is checked to a single directory.

When you have run a comparison, you can save the output to a file and view it in a standard application such as Excel.

This section discusses the following topics:

Snapshots

Snapshots are an advanced feature that define what a component should capture at the time it is installed and when it is analyzed for differences. Snapshots are largely a change management tool that helps in the process of detecting when a component installation has changed.

Because the source items for a component is deployed in the order in which they are listed in the component's XML file, it's possible for one source item to overwrite another that has already been deployed, if they both have the same name.

How Snapshots Help

One way that a snapshot is used is to validate that a deployment was done correctly. Because the source items for a component is deployed in the order in which they are listed in the component's XML file, it's possible for one source item to overwrite another that has already been deployed, if they both have the same name.

For example, you may have a component that installs the source for a component named pockets onto a host. One source (S1) of the pants component, pants/config.txt may be deployed, followed by a second, different source (S2), also pants/config.txt. If you were then to go and look at pants/config.txt you might think you were looking at S1, when in fact, config.txt now reflects the contents of S2. To avoid this sort of difficulty, you can define a snapshot that the provisioning software can use to make sure the correct baseline, post installation version of config.txt is used in any Model-to-install comparisons.

Another way snapshots are useful is when a component's source items might be transformed using the transform element, or otherwise changed during the course of deployment by an execNative call that you have defined. Without a snapshot of the "just installed" state, a Model-to-Install comparison would be made against the source items before they were deployed, which would lead to errors in the results.

It is not necessary to define a snapshot block, or specify that one be used, in your components. However, if one is not specified, the installed component that you are comparing against the model will contain no snapshot information, and will never present any differences.

Types of Comparisons and Their Features

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

- Model to Model (M-M)
- Model to Install (M-I)
- Install to Install (I-I)

Model to Model Comparisons

A Model to Model comparison is the simplest type of comparison. It compares the list of components that the provisioning software has installed on one host with the list of components that the provisioning software has installed on another host. This is

purely a comparison of the records that the provisioning software has about what it has installed. If there have been out of band changes made to either component list, they will not be detected. A Model-to-Model comparison takes place entirely on the Master Server based on information it has received about what has been deployed.

Model-to-Install Comparisons

A Model-to-Install comparison compares the 'just installed' state of the resources in a component or components to their current state on a given host or host set. You can specify that this comparison compares the model to hosts, host sets, or to component level.

A Model-to-Install comparison uses information about what components and subsequent resources have been deployed to a given host based on snapshot information. A snapshot is a record of the state of an installed component immediately after it has been deployed to a host. The component's XML model includes parameters that control the scope of this snapshot. For example, the XML model can list files, such as log files, that should be ignored during a comparison.

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

Unlike Model-to-Model comparisons, which compare one ideal installation to another, Model-to-Install comparisons compare the actual state of the installed component or components at deployment time against their current state.

Install-to-Install Comparisons

Install-to-Install comparisons compare the current states of the contents of directories in components that the provisioning software has installed on two different hosts. The list of elements (files, directories, and so on) from the reference host being compared in an Install-to-Install comparison are grouped into batches on the Master Server and then sent to the Remote Agent of the comparison host for processing. If you have defined some rules about file types or specific directories or files for the provisioning software to ignore, the Master Server will discard the appropriate items before sending the remainder of the list to the comparison host. The Remote Agents from each host must be set to use the same locale for Install-to-Install comparisons to function properly.

You can provide the provisioning software with scripts to be run before and after the comparison occurs, in case you need to manipulate files before running the comparison or need to clean up any files you have manipulated to prepare for the comparison.

Why Use Scripts in Install-to-Install Comparisons?

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

Using Scripts in Install-to-Install Comparisons

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

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

When you are performing an Install-to-Install comparison using the HTML user interface, you can select a checkbox labeled include temporary files generated by scripts during comparison. When you select this option, the page refreshes and provides two fields into which you can enter the prepare and cleanup blocks.

You can also specify alternate prepare/generate and cleanup blocks for the target host by un-selecting the appropriate boxes.

You can enter scripts such as these into the fields:

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

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

This cleanup block then removes the outputFile, `file.txt`, from the temp directory.

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

Running Comparisons

Time Synchronization for Comparisons

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

M-I (Model to Installation) Comparisons

An M-I comparison (sometimes called an M-I difference) is a comparison of a component model to a specific installation of a component. When relevant, this chapter's description of a resource type will include information about how the format or content of a resource type affects an M-I comparison.

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

M-I Comparison for IIS Types Introduction

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

M-I Comparison for COM+ Application

The COM+ Application is re-exported on the target system as an MSI file and compared against the MSI file used to install the application. M-I difference will only indicate that there were differences (i.e. the two binary files are different) but will not indicate the details of the differences.

M-I Comparison for COM Component

The COM component is compared as a binary file against the file used during installation. M-I difference will only indicate that there were differences (i.e. the two binary files are different) but will not indicate the details of the differences.

M-I Comparison for H Keys

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

M-I Comparison for Registry Settings

M-I Differencing is not supported for *.reg files. Snapshots will not be taking during installation resulting in nothing to difference during the M-I difference. If the user would like to difference registry changes they are encouraged to use the built-in Registry keys type.

M-I Comparison for MSI Files or Windows Installer Files

M-I Differencing is not supported for Windows Installer files. Snapshots will not be taking during installation resulting in nothing to difference during the M-I difference. Since ROX does not have first-hand knowledge of the actions taken during the installer run it is not feasible to determine what needs to be captured.

M-I Comparison for Windows Batch Files

M-I Differencing is not supported for Windows Batch files. Snapshots are not taken during installation resulting in nothing to difference during the M-I difference.

M-I Comparison for Comparison for Windows Scripting Host Files

M-I Differencing is not supported for WSH files. Snapshots will not be taking during installation resulting in nothing to difference during the M-I diff.

M-I Comparison for WebLogic WAR Files

Will use the standard file/directory MI difference approach.

M-I Comparison for WebLogic Web Application Settings

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

M-I Comparison for WebLogic JAR Files

Will use the standard file/directory MI difference approach.

M-I Comparison for WebLogic EJB Settings

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

M-I Comparison for WebLogic EAR Files

Will use the standard file/directory MI difference approach.

M-I Comparison for WebLogic Enterprise Application Settings

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

Comparisons User Interface

The Comparisons Page

The Comparisons user interface provides the functionality for users to create, run, and view the results of comparisons.

This page also contains controls for filtering the component list and selecting sort criteria. There are blue wedge shaped arrows over the comparison column that allow you to sort the list by and in either ascending or descending order.

Comparison	A user definable text string that names the comparison object. By default, the comparisons are listed in alphabetical order by component name. You can change the sort order between ascending and descending by clicking on one of the sort icons that is in the plans header area.
Description	An optional user defined text string that describes the comparison object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other comparison related pages to either create, view details for, or view the results of comparisons.
Actions for Checked Plans	From this area you can either delete or apply categories to the checked plan(s)

The Comparisons Details Page

The details page for comparisons is different for each of the three types of comparisons.

The Comparisons Details Page for M to M

Comparison	Displays the names the comparison object.	
Category	A user definable object that is used to filter the comparisons list. Categories are created using the categories page accessed from the Administrative page. Categories are then applied to a plan from the plans page. Categories can also be created by clicking in the "*Apply Categories . . .*" link.	
Description	An optional user defined text string that describes the comparison object. This attribute is not used by the provisioning software but can provide meaningful information to the user.	
Style of Comparison	This allows you to select the type of comparison you want.	
	Model to Model	Sets up the user interface to run a model to model comparison.
	Model to Install	Sets up the user interface to run a model to install comparison.
	Install to Install	Sets up the user interface to run an install to install comparison.

Reference Host	Specifies which hosts model to use as a reference model for the comparison.
Comparison Host	Specifies which hosts model to compare to the reference host's model.
Limits	This allows you to set a time limit for the provisioning software to complete the comparison. Some types of problems can cause the provisioning software to wait endlessly while attempting to take a snapshot of an installation. This feature provides a time-out to prevent the provisioning software from never finishing a comparison.
Buttons	By clicking one of these buttons you initiate one of the following functions.

Save	Saves the comparison using the current name.
Save As	Saves the comparison using a different name.
Save & Run	Saves the comparison using the current name and performs a comparison.
Previous Results	Displays previous results of a comparison.
Delete	Deletes the comparison.
Cancel	Takes you to the comparisons page without saving any changes.

The Comparisons Details Page for M to I

This comparison allows you to validate that current state of an installation by comparing it to its last know good state.

Because only the model reference and comparison object area changes, the following discussion will focus only on the area of the comparisons page that is different.

Scope This is a pull down menu that allows you to limit the scope of the comparison to either a host set, a host, or component.

Host Set - changes the following field on this page to allow you to select which host set to compare. The provisioning software then compares that last known state of the specified host set to its current state.

Host - changes the following field on this page to allow you to enter a host name to compare. The provisioning software then compares that last known state of the specified host to its current state.

Component - changes the following field on this page to allow you to enter a host name and component path to compare. The provisioning software then compares that last known state of the specified component to its current state.

Directories & Files to Ignore During Comparison
Allows you to specify things to ignore during this comparison. This feature allows the provisioning software to accommodate changes to things like log files without causing the comparison to fail.

The Comparisons Details Page for I to I

This comparison allows you to validate that current state of an installation by comparing it to an other know good installation.

Because only the model reference and comparison object area changes, the following discussion will focus only on the area of the comparisons page that is different.

Reference Host	Specifies which hosts to use as a reference model for the comparison. This also includes a second field that requires that you specify where in the hierarchical files structure to limit the comparison. If you use the top of the tree structure you will evaluate the entire host.
Comparison Host	Specifies which hosts to compare to the reference host's model. This also includes a second field that requires that you specify where in the hierarchical files structure to limit the comparison.

Options
Defines some parameters when taking a snapshot of the host.

Include All Sub-Directories in Comparison — This causes the provisioning software to recursively follow all subdirectories. within a parent directory.

Follow Symbolic Links Found in Directories — This causes the provisioning software to resolve symbolic links to determine that what the link points to is also valid.

Include Temporary Files Generated by Scripts During Comparison — This causes the provisioning software to include any legacy files generated by running a script. This can be used to verify that a particular script has or has not been run on all the hosts included in the comparison.

Directories & Files to Ignore During Comparison
Allows you to specify things to ignore during this comparison. This feature allows the provisioning software to accommodate changes to things like log files without causing the comparison to fail.

▼ How To View Comparisons List

Steps

- **In the left-hand navigation menu, click comparisons.**

  The HTML user interface displays the comparisons page, which lists the comparisons already checked in to the database. By clicking on one of the sort arrows in the header yo can change the sort criteria.

▼ How To View Comparisons Details

Steps

1. **In the left-hand navigation menu, click comparisons.**

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

2. **Find the row listing the plan you're interested in, and click details.**

   The HTML user interface displays the comparisons Details page for the comparison.

▼ How To Create an M to M Comparison

Steps

1. **In the left-hand navigation menu, click comparisons.**

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

2. **Find the row listing the plan you're interested in, and click details.**

   The HTML user interface displays the comparisons Details page for the comparison.

   On this page you select the type of comparison you want to define, and you define parameters for the comparison. When the page is first displayed, it is configured for a Model-to-Model comparison.

3. **In the style of comparison area, leave model to model selected.**

4. **In the reference host field, enter the name of the host whose model should serve as the reference (the standard) for the comparison. If you like, you can click select from list to perform a host search to find the host.**

5. **In the comparison host field, enter the name of the host whose model you would like to compare to the reference model. If you like, you can click select from list to perform a host search to find the host.**

6. **In the fields labeled limits, specify the maximum number of hours or minutes the comparison should run. By default, comparisons are allowed to run for 4**

hours.

7. **Click save to save the comparison.**

▼ How To Create an M to I Comparison

Steps

1. **In the left-hand navigation menu, click comparisons.**
   The HTML user interface displays the comparisons page, which lists the comparisons already checked in to the database.
2. **Find the row listing the plan you're interested in, and click details.**
   The HTML user interface displays the comparisons Details page for the plan.
3. **Click the button labeled model to install.**
   The HTML user interface displays the page, which now contains fields for defining a model-to-install comparison.
4. **Use the pull-down menu at the end of the model to install line to specify the contents that should be used in the comparison:**
   - directory and file properties Select this choice to confirm that both hosts have the same files and directories and that the files and directories are the same sizes.
   - file contents Select this choice if you want to determine if there are any differences in the contents of files on the two hosts.

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

   ---

5. **In the scope drop-down list, select whether you want to compare host or host set, or a component.**
6. **Specify the name of the host or host set.**
   If you have specified a scope of 'component', perform the following steps:
   a. **Specify the name of the host on which this component exists.**
   b. **Refresh the screen by clicking Apply change.**
   c. **Select the path to the component.**
7. **Set a time limit for the comparison using the limits controls.**

Once you've set a time limit, you can choose to save the comparison to run later or run it right now.

8. **Do one of the following:**
   - **To save the comparison without running it, click save.**
   - **To both save the comparison and run it right now, click save & run.**

▼ How To Create an I to I Comparison

Before You Begin

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

Steps

1. **In the left-hand navigation menu, click comparisons.**
   The HTML user interface displays the comparisons page, which lists the comparisons already checked in to the database.
2. **Find the row listing the plan you're interested in, and click details.**
   The HTML user interface displays the comparisons Details page for the comparison.
3. **Choose a level of detail for the comparison.**
   - Directory and File Properties — Are all the files and directories there? Are they the same sizes?
   - File Contents — Are there any text-level differences between files?
4. **Enter the name of the reference host and the path to the directory you want to compare against the target host.**
5. **Enter the name of the comparison host and the path to the directory you want to compare against the reference host.**
   The Include all subdirectories in comparison checkbox is checked by default.
   If you want the provisioning software to run a script before and/or after running this comparison, perform the following actions.
   a. **Check the Include temporary files generated by scripts during comparison checkbox.**
      The page will refresh and provide two fields for prepare (generate) and cleanup scripts:
   b. **Enter the scripts into the appropriate fields.**
      For information on scripts for prepare/generate and cleanup, refer to "Using Scripts in Install-to-Install Comparisons" on page 172.

c. **If you do *not* want the provisioning software to run the same scripts on the target (comparison) host as on the reference host, uncheck the appropriate boxes.**

- If you uncheck the 'generate' checkbox, a field is displayed so that you can provide an alternate prepare/generate script for the reference host.
- If you uncheck the 'cleanup' checkbox, a field is displayed so that you can provide an alternate cleanup script for the reference host.

If you want to specify resources to ignore during this comparison, proceed to "Choosing Directories or Files to Ignore During Comparisons" on page 182.

6. **Set a time limit for the comparison using the Limits controls.**

Once you've set a time limit, you can choose to save the comparison to run later or run it right now.

▼ Choosing Directories or Files to Ignore During Comparisons

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

To specify directories or files to be ignored, do the following:

Steps

1. **Enter the name of the directory or file you want to ignore, or provide an expression to represent the directories or files you want to ignore using wildcards (*).**

For example, if you want to ignore all log files, you can specify *.log to be excluded.

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

---

2. **Click add.**

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

The ignore box is checked by default to mark this directory or file so that it is not included in the system comparison.

If you want to remove this exclusion rule in a future comparison, you can uncheck the ignore box.

▼ Removing a Directory or File from the Ignore List

Steps

- **To remove a directory or file from the ignore list, click on remove.**

  You can also temporarily disable the ignoring of a directory or file by unchecking the ignore checkbox.

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

▼ Canceling a System Comparison in Progress

Steps

- **To cancel a system comparison currently in progress, click cancel.**

Note – If you cancel a comparison in progress, you are returned to the page listing all the comparisons. Occasionally, it may take a short while for the listing to refresh and show you that the comparison you canceled has been stopped. The reason for this is that some comparisons include file transfers between the Master Server and the comparison or reference hosts, and the comparison will not halt in mid-transfer, but will continue until the current file is completely transferred (refer to the description of each sort of comparison at the beginning of this chapter for specific information about files being transferred). The time it takes to refresh the page depends on the size of the file that provisioning software is currently transferring. When the comparison is completely halted, the updated list is displayed.

Running a Saved Comparison When a Component's Version has Changed

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

Summary of CLI Commands

Table 7–1 lists the CLI commands related to comparisons.

TABLE 7–1 Summary of cmp Commands

Command Name	Description
cmp.dj.add	Adds (starts) a new comparison.
cmp.dj.del	Deletes (stops) a new comparison.
cmp.dj.la	Lists running and completed comparisons.
cmp.dj.lo	Retrieves a running comparison to display its status.
cmp.ds.add	Adds comparison settings to the specified comparison.
cmp.ds.la	Lists all comparison settings.
cmp.ds.lo	Retrieves the specified comparison settings.
cmp.ds.sc	Associates a comparison settings with a set of categories.
cmp.ds.mod	Modifies the specified comparison settings.

CHAPTER 8

Configuring Notifications

You can configure the provisioning software to notify you or anyone with an email address when the following events occur:

- A plan starts to run, completes, or stops abnormally
- A comparison starts, completes, or stops abnormally

This section discusses the following topics:

Working with Notifications

You can be notified when any system, administrative, or custom event that you specify occurs. You can define a notification rule to apply to a single host or a host set, to an event containing a particular message, to be triggered by an event of a specified severity, or by an event of that severity or higher. You can specify what email address or addresses a notification will be sent to.

When you installed the provisioning software, you specified:

- the email address from which the provisioning software sends notifications
- the subject line that should appear in the header of notification email messages

The notification email address is set during installation of the provisioning software, and the default subject line text is "N1 Service Provisioning System Notification."

Notifications User Interface

The Notifications Page

The plans page list notifications, view notifications details, and create notifications.

Rule	A user definable text string that names the notification object. By default, the notifications are listed in alphabetical order by notification name.
Description	An optional user defined text string that describes the notification object. This attribute is not used by the provisioning software but can provide meaningful information to the user.
Action	Links that take you to other notification related pages to either create a new notification or view more information about a notification.
(Check Box)	Marks a notification for inclusion when one of the actions are clicked in the *Actions for Checked plans* area. When clicked, *Actions for Checked plans* delete all checked notifications.
Actions for Checked Plans	From this area you can delete notification(s).

The Notifications Details Page

The Notifications Details page allows you to view and edit a notification's information and attributes. Along with notification information and attributes this page contains buttons that allow you to save, save as, and delete a notification.

Rule
: A text filed that allows you to name an new notification or change the name of an existing notification.

Description
: A text filed that allows you to describe the notification object.

For Events meeting ALL of the Following Criteria
: This section of the notifications details page allows you to define the criteria for notifications. Within in this section there are two radio buttons (Event Type and Event Severity) that are mutually exclusive. By selecting on, you deselect the other.

Event Type
: Selecting this radio button enable the associated pull down menu that allows you to select from a list of events that will trigger a notification. The menu include the following events:

- Any Event
- Plan Starts

- Plan Ends Abnormally
- Plan Completes normally
- Comparison Starts
- Comparison Ends Abnormally
- Comparison Completes normally
- System Events
- Administrative Events
- Custom Events

Event Severity
Selecting this radio button enable the two associated pull down menus that allows you to select from a list of thresholds that will trigger a notification. The left menu include the following levels of severity.

- Information
- Warning
- Error

The right menu include the following logical operators that sets limits on the severity level.

- This Severity Only
- This Severity or Worse

Event Text Contains
A text filed that allows you specify the body of the email message.

Event Occurs
This section has two fields that determines which host or hosts will be monitored of the specified event.

On Host
A text box that allow you to specify which host to monitor for the specified event. If this field is left empty then you can specify a host set in the following field.

Or On Host Set
A pull down menu that allow you to specify which host set to monitor for the specified event. If this *on host* field is left empty then this field is evaluated.

Take The Following Action
A text box that allow you to specify one or more email addresses to send a notification when a triggering event is detected.

Buttons
By clicking one of these buttons you initiate one of the following functions.

Save	Saves the notification using the current name.
Save As	Saves the notification using the different user specified name.
Delete	Removes the notification from the database.
Cancel	Takes you to the notifications page without saving anything.

▼ How To View Notifications

Steps

- **In the left-hand navigation menu, click notification rules.**

  The HTML user interface displays the notifications page, which lists the notifications already saved in to the database.

▼ How To View Notifications Details

Steps

1. **In the left-hand navigation menu, click notification rules.**

   The HTML user interface displays the notifications page, which lists the notifications already saved in to the database.

2. **Find the row listing the notification you're interested in, and click details.**

   The HTML user interface displays a notifications Details page for the notification.

▼ How To Create Notifications

Steps

1. **In the left-hand navigation menu, click notification rules.**

   The HTML user interface displays the notifications page, which lists the notifications already saved in to the database.

2. **In the top row of the table listing rules, enter a name and a brief description for the rule you want to create, and click create.**

   The HTML user interface displays the Notification Rules Details page.

   Use the controls on this page to specify:

   - the event or events that will trigger the notification
   - the email address that the notification should be sent to.

3. **Select an event type from the Event Type drop-down list, or select a severity level.**

   Events

   You can create a notification rule for any of the following events:

   - any event that occurs
   - a plan starts
   - a plan ends abnormally
   - a plan completes normally
   - a comparison starts

- a comparison ends abnormally
- a comparison ends normally
- any system event occurs
- any administrative event occurs
- a custom event occurs

Severity Levels

The severity levels you can choose from are:

- Information
- Warning
- Error

You can specify whether you want the notification to be triggered when an event of a particular severity occurs, or whether an event of that severity or higher occurs. To determine what the severity of a particular type of event is, refer to Table 8–2.

4. **Choose This Severity Only or This Severity or Worse from the drop-down list.**

5. **If you want to select events based on the messages they automatically generate, you can specify part or all of an event text message in the event text contains field.**

   If you want to match a specific Event message, refer to the list of event messages in Table 8–1. You can use special character * to match any number of characters.

6. **If you want to specify that this rule applies only when events occur on a particular host or a particular host set, enter the name of a hosts or hosts in the on host field or select a host set from the on host set drop-down list. To specify multiple hosts by name, lists the names in the on host field and separate them with spaces, semicolons, or commas.**

7. **Enter an email address for the recipient of this notification.**

   You can specify multiple addresses by separating them with spaces, semicolons, or commas.

8. **When you have finished specifying the notification rule, click save.**

   The HTML user interface saves the notification rule, which will now be listed on the Notification Rules page.

Default Event Messages

The following are the default messages sent to the address you specified when you defined the notification rule.

TABLE 8–1 Default Event Messages

Event Type	Event Message
ROXErrorEvent	<always specified by the calling subsystem>
ROXWarningEvent	<always specified by the calling subsystem>
ROXInfoEvent	<always specified by the calling subsystem>
ROXComponentInstallEvent	During <preflight | deployment>, plan <plan name or id> is installing component <component name or id> on host <host name or id> on <datetime>.
ROXComponentUninstallEvent	During <preflight | deployment>, plan <plan name or id> is uninstalling component <component name or id> on host <host name or id> on <datetime>.
ROXComponentControlServiceEvent	During <preflight | deployment>, plan <plan name or id> is executing control service <control service name> for component <component name or id> on host <host name or id> on <datetime>.
ROXResourceInstallEvent	During <preflight | deployment>, plan <plan name or id> is installing resource <resource name or id> for component <component name or id> on host <host name or id> on <datetime>.
ROXResourceUninstallEvent	During <preflight | deployment>, plan <plan name or id> is uninstalling resource <resource name or id> for component <component name or id> on host <host name or id> on <datetime>.
ROXDifferenceAbortEvent	A comparison was cancelled on <datetime>. This comparison used settings: <difference settings or id>.
ROXDifferenceAbortRequestedEvent	A comparison was requested to be cancelled on <datetime>. This comparison used settings: <difference settings or id>.
ROXDifferenceCompleteEvent	A comparison completed on <datetime>. This comparison used settings: <difference settings or id>.
ROXDifferenceFailedEvent	A comparison failed <may contain “because <reason>”> on <datetime>. This comparison used settings: <difference settings or id>.

TABLE 8–1 Default Event Messages *(Continued)*

Event Type	Event Message
ROXDifferenceFoundEvent	A comparison found a difference <difference name or id> at <difference path> on <datetime>. This comparison used settings: <difference settings or id>.
ROXDifferenceProgressEvent	A comparison was <numeral percent> complete on <datetime>. This comparison is using settings: <difference settings or id>.
ROXDifferenceStartEvent	A comparison started on <datetime>. This comparison is using settings: <difference settings or id>.
ROXNodeStartEvent	<Master Server> started on host <host name or id> on <datetime>.
ROXNodeStopEvent	<Master Server> stopped on host <host name or id> on <datetime>.
ROXPreflightCompleteEvent	The preflight for plan <plan name or id> completed on <datetime>.
ROXTaskAbortEvent	During <preflight | deployment>, plan <plan name or id> <may contain: "on host <host name or id>"> was cancelled on <datetime>.
ROXTaskAbortRequestedEvent	Plan <plan name or id> cancellation was requested on <datetime>.
ROXTaskCompleteEvent	During <preflight | deployment>, plan <plan name or id> completed on <datetime>.
ROXTaskFailedEvent	During <preflight | deployment>, plan <plan name or id> failed <may contain: "because <reason>"> on <datetime>.
ROXTaskProgressEvent	During <preflight | deployment>, plan <plan name or id> was <percent complete> complete on <datetime>.
ROXTaskStartEvent	During <preflight | deployment>, plan <plan name or id> started on <datetime>.
ROXTaskTimeoutEvent	During <preflight | deployment>, plan <plan name or id> timed out <may contain: "on host <host name or id>"> on <datetime>.
ROXStepCustomStepEvent	<always specified by the calling subsystem>
ROXStepCompleteEvent	During <preflight | deployment>, step <step name> in plan <plan name or id> completed on host <hostname or id> on <datetime>.

TABLE 8–1 Default Event Messages *(Continued)*

Event Type	Event Message
ROXStepFailedEvent	During <preflight | deployment>, step <step name> in plan <plan name or id> failed on host <hostname or id> <may contain: "because <reason>"> on <datetime>.
ROXStepProgressEvent	During <preflight | deployment>, step <step name> in plan <plan name or id> was <percent complete> on host <hostname or id> on <datetime>.
ROXStepStartEvent	During <preflight | deployment>, step <step name> in plan <plan name or id> started on host <hostname or id> on <datetime>.
ROXUserLoginEvent	User <user name or id> logged in on <datetime>.
ROXUserLoginFailedEvent	User <username> failed to log in <may contain: "because <reason>"> on <datetime>.
ROXUserLogoutEvent	User <username or id> logged out on <datetime>.

Event Severities

For events, the following strings and event severities apply:

TABLE 8–2 Event Severities

Event	Event Description	Severity
ROXErrorEvent	System Error	ERROR
ROXWarningEvent	System Warning	WARNING
ROXInfoEvent	System Information	INFO
ROXComponentInstallEvent	Component Installed	INFO
ROXComponentUninstallEvent	Component Uninstalled	INFO
ROXComponentControlServiceEvent	Component Control Service Executed	INFO
ROXResourceInstallEvent	Resource Installed	INFO

TABLE 8–2 Event Severities *(Continued)*

Event	Event Description	Severity
ROXResourceUninstallEvent	Resource Uninstalled	INFO
ROXDifferenceAbortEvent	Comparison Canceled	INFO
ROXDifferenceAbortRequestedEvent	Comparison Cancellation Requested	INFO
ROXDifferenceCompleteEvent	Comparison Completed	INFO
ROXDifferenceFailedEvent	Comparison Failed	ERROR
ROXDifferenceFoundEvent	Difference Found	WARNING
ROXDifferenceStartEvent	Comparison Started	INFO
ROXNodeStartEvent	Node Started	INFO
ROXNodeStopEvent	Node Stopped	INFO
ROXPreflightCompleteEvent	Preflight Completed	INFO
ROXTaskAbortEvent	Plan/Preflight Canceled	INFO
ROXTaskAbortRequestedEvent	Plan/Preflight Cancellation Requested	INFO
ROXTaskCompleteEvent	Plan/Preflight Completed	INFO
ROXTaskFailed	Plan/Preflight Failed	ERROR
ROXTaskStartEvent	Plan/Preflight Started	INFO
ROXStepCustomStepEvent	Custom Event Sent	INFO
ROXStepCompleteEvent	Step Completed	INFO
ROXStepFailedEvent	Step Failed	ERROR
ROXStepStartEvent	Step Started	INFO
ROXUserLoginEvent	User Logged In	INFO
ROXUserLoginFailedEvent	User Failed to Log In	INFO
ROXUserLogoutEvent	User Logged Out	INFO

Summary of Notification CLI Commands

Table 8–3 summarizes the CLI commands for managing rules for notifications.

TABLE 8–3 Summary of the rule Commands

Command	Description
rule.add	Adds a notification rule.
rule.del	Deletes a notification rule.
rule.la	Lists all notification rules.
rule.lo	Retrieves a specific rule.
rule.mod	Modifies an existing rule.

APPENDIX **A**

Special Names

Reserved Names

Within the provisioning software there are certain names that are reserved for system use. If you attempt to use one of the reserved names, the provisioning software will give you an error message regarding the name.

- Paths

  /system — The provisioning software reserves the */system* path and everything under it.

- Hosts

  masterserver — This is the Master Server.

- Host Types

  crhost — This the default host type.

- Component Variables

name	Name of the component
description	Description of the component
label	Label of the component
softwareVendor	Software vendor of the component definition
author	Author of the component definition

- Host Types

name	Name of the target host
description	Description of the target host
hostTypeName	The name of the host type of the target host

raIP	IP address of the remote agent on the target host
raPort	Port of the remote agent on the target host
raHomeDir	The absolute path of the remote agent home directory on the target host
raDataDir	The absolute path of the remote agent data directory on the target host
raTmpDir	The absolute path of the remote agent temporary directory on the target host
raConfigDir	The absolute path of the remote agent config directory on the target host

Glossary

Component model: An XML representation of a component. The component model is a bill of materials of the applications that includes:

- A list of resources used by the application
- Install steps
- Uninstall steps
- Dependencies

Component: A component is a unit that is deployed to hosts in the provisioning software. It comprises a build between a component model and a specific set of resources for an application, such as a custom Java application or an infrastructure software package. A component can comprise multiple versions of an application. In this case, each checked-in version of the application is represented by a separate component build.

Deployment: The execution of a plan, which automatically distributes a component or components included in the plan. The deployment follows installation instructions set in the plan. Deployment is always preceded by a preflight. If the preflight is successful, the Plan Executor will actually deploy/install components to the target host.

Comparisons: A provisioning software feature that searches for and identifies differences between hosts. There are three types of comparisons: Model to Install, Model-to-Model, Directory-to-Directory.

The *install-to-install comparison* examines the contents of two hosts' file systems and reports any differences.

The *model-to-install comparison* compares what the Master Server reports is installed on a host against what is actually installed on the host, and reports any differences.

	A *model-to-model comparison* examines the deployment repository and history (stored on the Master Server) for two hosts, and reports any differences.
execNative:	An optional call out to custom scripts from a plan XML.
Gold server:	A reference (outside) host that contains files, directories, and other resources comprising an application you want to deploy. These resources are checked in (uploaded) to the provisioning software from the gold server.
Host:	A server connected to the provisioning software. A host can represent an individual server or a set of servers.
Host set:	A user-defined, logical grouping of hosts sharing one or more common attributes, such as physical location or functional group. A defined host set allows quick and easy application updates to all hosts in the set, as well as install-to-install comparisons between the hosts. Host sets can be nested.
Host type:	A base class of servers bound by a set of common attributes, all of which are user-defined. Hosts belonging to a host type set use particular attribute values in variable substitutions in all deployments. Host types can be used to categorize hosts into logical groupings, such as sets of applications required for the hosts, geographic locations of the hosts, or functional groups. Host types also facilitate host searches by allowing users to search for all hosts of a certain type.
Host search:	A query run on the repository that yields a list of hosts whose attributes match those specified in the query. You can use host searches to create lists of hosts that have the same host type, are running the same applications, are configured with the same subnet masks, and so on.
Local Distributor, or LD	An engine, installed on a machine with the provisioning software, that acts as a link between other objects, such as the Master Server to Remote Agents, the Master Server to other LDs, and another LD to Remote Agents. Designed to maximize bandwidth efficiency and speed, LDs can also provide secure network connections for navigating restricted environments.
Master Server, or MS:	The main processing engine in the provisioning software. It is installed on a dedicated machine at a central location, where it can connect to any of the data center environments. The MS provides the centralized data storage, data processing, and user interfaces.
Plan:	An XML representation of the deployment script that references components to be deployed. A plans include: ■ Install/uninstall steps. ■ Start/Stop components (optional). ■ Verification steps (optional)

- ExecNatives (optional)

Plan executor: The provisioning software's deployment engine. It handles preflights and actual deployments.

Preflight: A test "deployment" of a plan to a simulated UNIX environment that:

- Checks availability and connectivity of target hosts.
- Confirms permissions.
- Validates dependencies.

A preflight finds and reports any errors or potential errors that could affect the actual deployment. Preflights always automatically proceed a deployment; they can also be run as standalone procedures that are not automatically followed by a deployment.

Remote Agent, or RA: An engine installed on any machine in the provisioning software to which components are deployed. On that machine, the RA manages tasks such as installing software, controlling services, collecting information to send back to the MS.

Resources: Files that are deployed to hosts when a plan is executed.

Resource types: Built-in resource categories identified and supported by the provisioning software. When a resource is checked in, it must be tagged with the appropriate resource type. Resource types include files, directories, IIS Types, and COM+ components.

Snapshot: A capture of resources stored on a Remote Agent during a deployment. The snapshot is used during comparisons between hosts (install-to-install) or between a host and the model of the host on the Master Server (model-to-install).

Verification step: An optional URL or process test that can be run at the end of a deployment to ensure successful completion of the plan.

Index

S

V