Product Brief

Sun™ ONE Meta-Directory

Version 5.1.1

817-3895-10
May 2004

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

Contents

Preface

The Meta-Directory Product Brief describes key concepts of Sun ONE™ Meta-Directory software. This preface contains the following sections:

- What You are Expected to Know
- The Sun ONE Meta-Directory Documentation Set
- Organization of This Guide
- Documentation Conventions
- Where to Find Additional Information
- Your Feedback on the Documentation

What You are Expected to Know

This *Product Brief* is intended for use by system administrators. Before reading this guide, you should be familiar with Lightweight Directory Access Protocol (LDAP) as well as the Windows NT or the Solaris Operating System on the servers on which Meta-Directory™ is installed.

The Sun ONE Meta-Directory Documentation Set

The Sun ONE Meta-Directory 5.1.1 documentation set includes the following materials:

- *Sun ONE Meta-Directory Deployment Guide* describes the Sun ONE Meta-Directory software and how to plan and implement a Sun ONE Meta-Directory software solution.

- *Sun ONE Meta-Directory Installation Guide* gives instructions on how to install the Sun ONE Meta-Directory software on both Solaris™ and Windows NT systems.
- ***Sun ONE Meta-Directory Product Brief*** (this guide) documents key concepts of the Sun ONE Meta-Directory.
- *Sun ONE Meta-Directory Administration Guide* documents how to set up and manage the Sun ONE Meta-Directory software. Configuring the Join Engine and components as well as managing the system from the Console and the command-line are addressed.
- *Release Notes* include information about what is new in this release, last minute installation tips, known software limitations, and how to report problems.

NOTE Be sure to check the Meta-Directory documentation web site for updates to the release notes and for revisions of guides. Updated documents will be marked with the revision date.

`http://docs.sun.com/prod/S1_MetaDir_511`

Organization of This Guide

The table below lists and briefly describes the content of the *Product Brief*:

Table 1 *Product Brief* Chapters

Chapter	Description
"Preface"	An outline of the Product Brief and a description of the Sun ONE Meta-Directory documentation set
Chapter 1, "Meta-Directory Overview"	A brief explanation of Meta-Directory concepts
Chapter 2, "Quick Start"	An easy reference on the steps needed to set up Meta-Directory
Chapter 3, "Meta-Directory Principles at Work"	An example of a typical way to configure Meta-Directory
Chapter 4, "Console Interface"	An overview of the different Sun ONE product consoles used within Meta-Directory

Documentation Conventions

In all Sun ONE Meta-Directory documentation, certain typographic conventions and terminology are used to simplify discussion and to help better understand the material.

Typographic Conventions

This book uses the following typographic conventions:

- *Italic type* is used within text for book titles, new terminology, emphasis, and words used in the literal sense.
- `Monospace font` is used for sample code and code listings, API and language elements (such as function names and class names), filenames, pathnames, directory names, HTML tags, and any text that must be typed on the screen.
- *Italic serif font* is used within code and code fragments to indicate variable placeholders. For example, the following command uses *filename* as a variable placeholder for an argument to the gunzip command:

```
gunzip -d filename .tar.gz
```

Terminology

Below is a list of the general terms that are used in the Sun ONE Meta-Directory documentation set:

- *Meta-Directory* refers to Sun ONE Meta-Directory software and any installed instances of the Sun ONE Meta-Directory software.
- *Meta-Directory components* refers to the collective set of Sun ONE Meta-Directory™ components and software you have installed and running on your system, including the Join Engine and any external data source connectors.
- E*xternal data source* refers to any user data that originates outside of the core Meta-Directory components, whether the data is coming from another database, directory server, data file, or other source of data.

- Directory Server refers to an installed instance of Sun ONE Directory Server™, iPlanet Directory Server and Netscape Directory Server.

  The Sun ONE Meta-Directory software can synchronize data using any LDAPv2 and LDAPv3-compliant directory server, as long as the LDAP server supports a change log mechanism similar to the one implemented in Netscape Directory Server 4.1*x*. The term *Directory Server* refers to the instances of iPlanet Directory Server and Netscape Directory Server that you have installed to work with iPlanet Meta-Directory.

- Similarly the term *Administration Server* refers to an installed instance of Sun ONE Administration Server, whether it be used with the Meta-Directory components or another Sun ONE server.

- *NETSITE_ROOT* is a variable placeholder for the home directory where you have installed the Sun ONE Meta-Directory software and any other Sun ONE servers installed into the same server group.

- The term *flow* is used rather loosely to refer to the process of synchronizing data between an external data source and the Meta View. You ‘flow’ data through a connector to the Connector View and then ‘flow’ it to the Meta View. The contrary is also true, you ‘flow’ data from the Meta View back to the Connector Views and out to the external data sources.

Where to Find Additional Information

In addition to the Sun ONE Meta-Directory documentation set, you should be familiar with the documentation for products that are used in conjunction with it. Of particular interest are the Sun ONE Console and Sun ONE Directory Server documentation set. This section lists additional sources of information you may find helpful as you use the Sun ONE Meta-Directory software.

Sun ONE Console Documentation

You can find the Sun ONE Console documentation at the following site:

`http://docs.sun.com/db/prod/s1.ipconsole.2`

Sun ONE Directory Server Documentation

You can find the Sun ONE Directory Server documentation at the following site:

`http://docs.sun.com/db/prod/s1.ipdirs`

Directory Server Developer Information

In addition to the Directory Server documentation, you can find information on Meta-Directory, LDAP, the iPlanet Directory Server, and associated technologies at the following Sun ONE developer sites:

`http://www.sun.com/developers/`

Other Sun ONE Product Documentation

Documentation for all Sun ONE servers and technologies can be found at the following web site:

`http://docs.sun.com/db/prod/sunone`

Sun ONE Technical Support

You can contact Sun ONE Technical Support through the following location:

`http://www.sun.com/support/`

Your Feedback on the Documentation

Sun Microsystems is interested in improving this documentation and welcomes your comments and suggestions. Use the following web-based form to provide feedback to us:

`http://www.sun.com/hwdocs/feedback/`

Please provide the full document title and part number in the appropriate fields. The part number can be found on the title page of the book or at the top of the document, and is usually a seven or nine digit number. For example, the part number of the *Sun ONE Meta-Directory Installation Guide* is 817-3896-10.

Chapter 1

Meta-Directory Overview

This chapter provides an overview of the Sun ONE Meta-Directory software including how data is integrated and how components interact. It contains the following sections:

What is Meta-Directory?

Sun ONE Meta-Directory is a set of software components that synchronize data from one or more external data sources into a single repository. This repository then becomes the authoritative source for the combined data, acting as a starting point for searches and for modifications that flow back to the original data sources. These data sources can include LDAP directories, Oracle databases, and other proprietary formats.

Meta-Directory Components

The Join Engine and direct and indirect connectors are the components which flow data into and out of Meta-Directory and integrate it into one definitive source. The connectors transfer information to and from a number of data sources to the Join Engine. The Join Engine links these entries together to form one repository.

Connectors

A *connector* transfers information between an external data source and its corresponding *Connector View*. When configuring a connector to transfer the information, an LDAP sub-tree is created on a Directory Server. This sub-tree is called a Connector View. The Connector View is populated with LDAP-structured copies of the entries that reside in the external data source. From this Connector View, Meta-Directory is able to integrate the new LDAP-structured data into an LDAP Meta View.

There are two types of connectors: direct and indirect. Direct connectors are plugins to the join-engine, which allow bi-directional synchronization of data from SunONE Directory Server (Sun One Directory Server 5.2, iPlanet Directory Server 5.x, and Netscape Directory Server 4.16) and Oracle database server (8.1.7, 8.1.9, and 9i) to the Meta View stored in SunONE Directory Server. Indirect connectors allow bi-directional synchronization of data from other external data sources (such as, Active Directory Server, NT Domain, Formatted-Text-Data, Novell Directory Server, and Lotus Notes) to the corresponding Connector View stored in the SunONE Directory Server.

Direct Connectors

SunONE Directory Server (Sun One Directory Server 5.2, iPlanet Directory Server 5.x, and Netscape Directory Server 4.16) an Oracle database accessible using Oracle Call Interface have direct connections to the Join Engine. In order to read and write to an entry stored within Oracle’s SQL database, the Join Engine uses the *Database connector* plug-in to provide direct, two-way access. (Because the Database Connector is a Join Engine plug-in as opposed to a connector outside the Join Engine, it is considered a direct connector.)

Indirect Connectors

Indirect connectors are used to transport entries stored in external data sources that use protocol not directly accessible by the Join Engine. The indirect connectors manage the process, transforming the data using rules and filters. (The indirect connector rules include Attribute Flow Rules, Default Attribute Value Rules and Filter Rules.) Meta-Directory supports the following indirect connectors:

Universal Connector and Universal Text Parser

The Universal connector (also known as the Universal Text Connector or UTC) is an indirect connector that enables the transfer of data between data sources and a Connector View. The Universal Text Parser (UTP) is a set of text file parsers and generators that are used with the UTC to make certain text files [currently Comma-Separated Values (CSV) files, LDAP Data Interchange Format (LDIF) files and Name-Value Pair (NVP) files] compatible with the Connector View.

NT Domain Connector

The NT Domain connector is a Universal connector with Windows NT-specific Perl scripts and binaries that provides two-way synchronization of user and group data between a Windows NT SAM database and its Connector View.

Active Directory and Exchange Connectors

The Active Directory connector is a Universal connector with Active Directory-specific Perl scripts and binaries that provides two-way synchronization of user and group data between an Active Directory database and its Connector View.

Novell and Notes Connectors

The Novell Directory and Lotus Notes Connectors have been implemented as indirect connectors. However, unlike the existing Indirect connectors such as - Universal Text Parser, Active Directory, Microvolt Exchange and the NT Domain connectors, this connector is not UTC-based. It is based on a new connector framework introduced in V5.1. This new connector framework uses an intermediate Masque database to perform change detection and loop detection for data in Novell Directory Server or Lotus Notes.

Join Engine

The core service of Meta-Directory is the Join Engine, responsible for linking entries and controlling the flow of information from a Connector View into or out of the *Meta View*. The Meta View is an LDAP sub-tree where entries from one or more Connector Views are linked, stored and accessed. The Join Engine synchronizes the data by flowing it through a sequence of rules and filters which *map* attributes from the Connector View with corresponding attributes in the Meta View. (*Mapping* is the assignment of an attribute or entry in one source to link with a particular attribute or entry in another source.) The movement of this data through the sequence of rules is referred to as the *join process*. The Join Engine also monitors the Connector Views for changes and incorporates them into the Meta View; conversely, it monitors the Meta View for changes and incorporates them into the Connector View and back to the related external data source.

Join Process

The join process is the movement of data through a sequence of rules and filters that are used by the Join Engine to determine how Connector View entries will be represented in the Meta View. To successfully join entries, the Join Engine must match data in the Connector View with a congruous entry in the Meta View. To do this, rules containing values and attributes are issued against the Connector View. These rules find a matching entry, and create a link to it, in the Meta View. The search strings include Join Rules, Attribute Construction Definitions, Attribute Flow Rules, Filters, and DN Mapping Rules.

Meta-Directory Views

In connecting data from several data sources, Meta-Directory allows the information to be accessed in two ways:

- The Meta View, a representation of the culmination of the join process — a view of the combined data from all Connector Views.
- The Connector View, a representation of the data from its initial source, the external data source.

The following sections describe these views in more detail.

Meta View

The *Meta View* is a unified view of LDAP entries from one or more Connector Views, representing the result of the Join Engine’s join process. After the Join Engine receives information from a Connector View, it synchronizes the information in the Meta View.

From the Meta View, you can view the linked entries as well as modify them and send the modifications back to the original Connector Views. Meta-Directory supports only one Meta View per Join Engine.

Connector Views

A *Connector View* is an LDAP representation of an entry (or entries) that resides in an external data source. This representation is merged with other Connector Views to form the Meta View. In order for this merger to take effect, you must specify that a Connector View is a *participating view*. Adding a participating view allows the Connector View to 'participate' in the join process; enabling it allows the data to flow from the Connector View into the Meta View. (It is through the participating view that the rules of the join process are applied.)

How Meta-Directory Works

Information from data sources is channeled to one or more Connector Views via a direct or indirect connector. The Join Engine combines the data from the Connector Views into one Meta View. All the information in the Meta View is then accessible via a directory server and other web-based applications. This is how the process plays itself out:

1. The Join Engine works with direct and indirect connectors that channel data from external sources into a Directory Server Connector View.

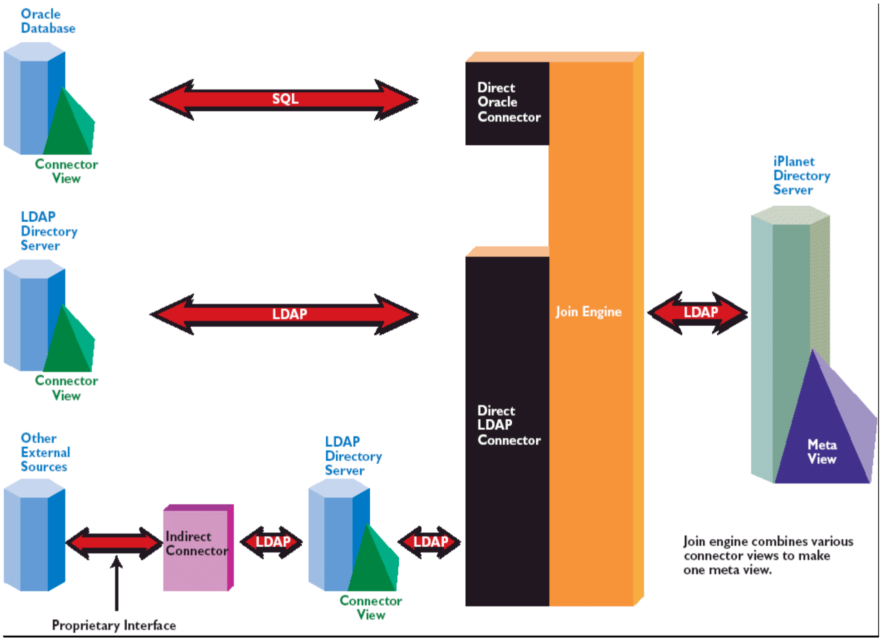
   a. The direct connectors read entries from either an LDAP directory or an Oracle database and map the data into LDAP-configured Connector Views.

   b. For other sources of data, an indirect connector transforms information from a proprietary format into LDAP and flows this data into an LDAP-configured Connector View.

   The data in these LDAP-configured Connector Views can then be flowed into, and stored on, a Meta View (another directory subtree in a Directory Information Tree) via the direct LDAP connector.

2. The Meta View becomes the replicated source for the combined data, allowing searches and changes to be made and flowed back to the original source.

Figure 1-1 Data Integration into Meta-Directory:



Chapter 2

Quick Start

Follow these general steps to get Meta-Directory up and running.

1. Install Meta-Directory and its components.

   Meta-Directory is used to establish an interdependent relationship between two or more sources of data, allowing the data to be integrated, viewed, searched and/or modified from one authoritative source. The components used to join the data include the Join Engine and connectors (both direct and indirect). During installation, you will be asked to choose a Directory Server and a path to a directory on the Directory Server where the Meta-Directory configuration will be stored. For more information on installation, see the *Sun ONE Meta-Directory Installation Guide*.

2. Start Sun ONE Console, your server group's Administration Server and your external data source.

   Sun ONE Administration Services, that includes Sun ONE Administration Server and Sun ONE Console, was installed along with Meta-Directory. Sun ONE Administration Server (accessed through Administration Server Console) is the common front-end for all Sun ONE Servers. Sun ONE Console manages all installed Sun ONE server groups. For instance, you might install Directory Server in one directory tree and Meta-Directory in another. Each server group will have its own Administration Server but, Sun ONE Console allows the management of both server groups from a single graphical user interface. Similarly, you access Meta-Directory from Sun ONE Console although the initial Meta-Directory configuration (Step 3 and Step 4) is achieved through Sun ONE Console. For more information on Console windows and applications, see Chapter 4, "Console Interface" or the Sun ONE Console and Administration Server documentation set.

3. Create an instance of the Join Engine.

   The Join Engine manages the flow and integration of data from different sources into one Meta View. An instance of the Join Engine needs to be configured to create the Meta View. The URL of the Directory Server where the Meta View will be located as well as an authenticating DN and password are needed for instance creation. Once the instance is created, a Meta View is created automatically in the navigation tree of Meta-Directory Console. The instance is created from Sun ONE Console and Meta-Directory Console can be opened from the instance in Sun ONE Console. For more information on the Join Engine, see Chapter 5, "Configuring the Join Engine" of the *Administration Guide.*

4. Connect one or more sources of external data to Meta-Directory by creating one or more instances of a server.

   An indirect or direct connector is used to flow data from the external data source to the Connector View. Connectors allow information to flow in two directions: original or modified data can move from the data source to the Connector View and modified data can flow from the Connector View back to the data source. You associate external data with a Connector View in one of two ways:

   a. Create an instance of an indirect connector, configure and apply the indirect connector rules and start the instance.

      Any external data source that goes through the process of having its data transformed into LDAP *before* flowing through the Join Engine is considered to have an indirect connection to the Join Engine. Active Directory, NT Domain and Universal Text Parser are indirect connectors which translate proprietary data into LDAP. When an instance of one of these indirect connectors is created (from Sun ONE Console, not Meta-Directory Console), it links to an external data source and creates a Connector View on a Directory Server with which to view the data. Starting the indirect connector instance begins the flow of data from the external data source to the Connector View. (It is important to remember that creation of the instance does not flow data; it only opens a data path.)

      Meta-Directory provides rules to manage the flow of data from the external data source through the indirect connector and to the Connector View. These *indirect connector rules* can alter the flow in three ways: by filtering entries from the process, by setting entry ownership, and by matching attributes in the data source to attributes in the Connector View. Other indirect connector configuration choices include setting up log options, scheduling synchronization, choosing from available attributes, and, in the case of the Universal connector only, adapting the connector with a Perl script to help it to synchronize a proprietary data source. Indirect connector rules are configured at the connector level on the navigation tree but are applied to specific connectors separately.

b. Create a direct Connector View and start it.

   The Join Engine allows access to LDAP Directory Servers and Oracle databases within its architecture; these data sources are considered to have a direct connection. A direct connector is not needed; only a Connector View need be created. (As there is no connector, the indirect connector rules do not apply either.)

   To create a direct Connector View, a Directory Server or an Oracle database needs to be recognized by the Join Engine. You configure a Directory Server by inputting address information (host name, port, DN name and password), scheduling operation and data modification times and setting up binary attributes. An Oracle database is also configured with database address information with the addition of changelog scripts, operation and data modification times and column selections. Once the Data Servers are connected to Meta-Directory, the direct Connector View can be instantiated by pointing it to the server.

5. Define the join process rules.

   The key to managing the manipulation of data from the Connector View to the Meta View is the join process. This process is the flow of data through the Join Engine and the data's synchronization within the Meta View. Administrators write join process rules to take full advantage of Meta-Directory's synchronization tools. There are five types of rules: join rules, attribute flow rules, DN mapping rules, filters and attribute construction rules. (Don't confuse these rules with the similarly-named indirect connector rules which manage the flow of data between the external data source and the Connector View.) Generally, these rules decide which entries (or attributes) from the Connector View will exist in the Meta View. Specifically, the administrator can, among other things, prevent specific entries from flowing back and forth and specify which attribute values in the Connector View map to which attribute values in the Meta View.

   The join process rules written at this step will not actually manage the data until they are specifically applied to a participating view in Step 6.

6. Configure a participating view and apply the join process rules.

   A participating view must be added before data will flow from the Connector View into the Meta View. Once added and enabled, the participating view is telling the Join Engine that the corresponding Connector View is ‘participating’ in the data flow. Applying the join process rules to the participating view will manage data flow between the Connector View and Meta View.

   The previously-configured join process rules are applied to the participating view after it has been added to the navigation tree. If no rules have been configured, the participating view uses *Atomic* rules. (The Atomic rules are a set of default DN Mapping rules and Attribute Flow rules that allow data to flow in the absence of previously-configured join process rules.) In addition to the join process rules, the direction of data flow, a data refresh schedule and data refresh filters can be applied to a participating view.

7. Start the Join Engine.

   The Join Engine is started from either Sun ONE Console or Meta-Directory Console. Even when the Join Engine is started, data will not flow until the participating view and the Connector View are enabled.

8. Enable the participating view.

   The participating view is enabled from the Join Engine’s Operations window in Meta-Directory Console. When you enable the participating view, its Connector View will flow data into the Meta View, using the join process rules applied to it.

9. Ensure that the Connector View is enabled.

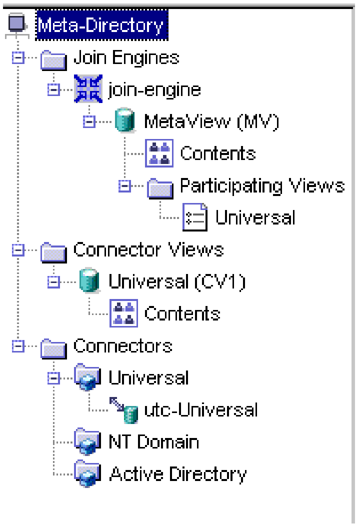
   By default, the Connector View is always enabled so data flow from the external source to the Meta View is always possible, providing the prior eight steps have been executed correctly. The Connector View’s status can be verified from the connector’s Operations window in Meta-Directory Console.

10. Refresh the data.

    Refreshing the data ensures that the most recent data is flowing. This step is most important when changes and modifications have been made to data.

    Shows how the Meta-Directory navigation tree appears if you were to create a Universal connector and its corresponding Connector View, then enabled it as a participating view.

Figure 2-1 Meta-Directory Navigation Tree With a Configured Connector View



Chapter 3

Meta-Directory Principles at Work

This chapter uses an example to show how to configure Meta-Directory to integrate three different external data sources into one unified Meta View. In order to make this example compliant with the Windows NT or Solaris operating system, the sample uses data sources that work with both. These include an LDIF data file imported into a Directory Server, a CSV data file using the Universal Text Parser (UTP), and an Oracle database using the Database Connector. These data samples are small but the principles of working with them using Meta-Directory are the same as they would be for large bodies of data. This chapter includes the following sections:

- LDIF Data File
- CSV Data File
- Oracle Database
- Background Information
- Configuration of Connectors and Join Process Rules
- Integration of the Data With Join Rules

LDIF Data File

An LDIF file is a text version of LDAP data. The following LDIF data is what will be imported into Directory Server. Once it is imported, the LDAP data sub-tree is directly accessible by Meta-Directory and would be considered a direct connection.

Code Example 3-1 LDIF Data

```
dn: uid=aquinn,o=firstgroup
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetorgperson
cn: Al Quinn
sn: Quinn
mail: alanquinn@siroe.com
title: Writer
uid: aquinn
givenname:Alan

dn: uid=bmoore,o=firstgroup
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetorgperson
cn: Betty Moore
sn: Moore
mail: bettymoore@siroe.com
title: Designer
uid: bmoore
givenname: Betty

dn: uid=cwayne,o=firstgroup
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetorgperson
cn: Cal Wayne
sn: Wayne
mail: calvinwayne@siroe.com
title: Accountant
uid: cwayne
givenname: Calvin

dn: uid=dgordon,o=firstgroup
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetorgperson
cn: Debby Gordon
sn: Gordon
mail: deborahgordon@siroe.com
title: Writer
uid: dgordon
givenname: Deborah

dn: uid=ecrane,o=firstgroup
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetorgperson
```

Code Example 3-1 LDIF Data *(Continued)*

```
cn: Eric Crane
sn: Crane
mail: ericcrane@siroe.com
title: Designer
uid: ecrane
givenname: Eric
```

CSV Data File

A CSV (comma-separated values) file is a text file format that can be output by a standard spreadsheet program. (Its name is derived from the fact that each record is stored in a single line with the individual items of data separated by commas.) The following CSV data will be imported into Meta-Directory via the UTP. (The UTP is considered an indirect connector as it is necessary to map the CSV file data to the Directory Server's LDAP schema before it can be read by Meta-Directory.)

Code Example 3-2 CSV Data

```
REALNAME,LASTNAME,EMAIL,PHONE,BUILDING,JOBTITLE,ALTERNATIVENAME,NICKNAM
E

Frankie Shaden,Shaden,frankshaden@siroe.com,408 555
1234,15,Accountant,fshaden,Frank

Gary Parker,Parker,garyparker@siroe.com,408 555
5678,7,Writer,gparker,Gary

Hal Kent,Kent,haroldkent@siroe.com,408 555 9012,18,Designer,hkent,Harold

Indy Connor,Connor,indranilconnor@siroe.com,408 555
3456,7,Accountant,iconnor,Indranil

Joe Hayes,Hayes,josephhayes@siroe.com,408 555
7890,15,Writer,jhayes,Joseph
```

Oracle Database

Data stored in an Oracle database can be viewed in Meta-Directory with a Database Connector. The following data will be accessed from an Oracle 8 database via Structured Query Language (SQL). The Database Connector is considered a direct connector because it plugs directly into the Join Engine.

Table 3-1 Oracle Data

Employee ID	First Name	Last Name	Salary	Department ID
aquinn	Al	Quinn	75,000	10
bmoore	Betty	Moore	85,000	43
cwayne	Cal	Wayne	80,000	24
dgordon	Debby	Gordon	95,000	10
ecrane	Eric	Crane	85,000	43
fshaden	Frankie	Shaden	80,000	24
gparker	Gary	Parker	95,000	10
hkent	Hal	Kent	85,000	43
iconnor	Indy	Connor	80,000	24
jhayes	Joe	Hayes	95,000	10

Background Information

Siroe, Inc. is a small company that outsources services. It currently has ten employees on its payroll. Personal and employment information on their employees is divided among three data sources. Siroe's Directory Server contains names and email addresses of five employees. Similar information for five consultants is in an Excel spreadsheet program. An Oracle database contains the salary information, employee numbers, and departments of all ten employees. Siroe needs to integrate all three of these sources and identify the employees by job function: four writers, three designers, and three accountants.

The example assumes that iPlanet Directory Server and the Sun ONE Meta-Directory software have been installed and a Join Engine has been created.

Configuration of Connectors and Join Process Rules

Siroe, Inc. has hired a systems administrator, Tracy, to help with the integration of their data. Tracy decides to flow the data on Siroe's Directory Server first. The next steps are integrating the CSV file exported from Excel and then syncing the Oracle tables into Siroe's unified employee database.

Flowing LDIF Data to the Connector View

Siroe has two directory servers: one is an iPlanet Directory Server and the other is Netscape Directory Server 1.0. The Directory Server 1.0 is not completely compatible with Meta-Directory so Tracy exports its LDAP data to an LDIF file in order to import it into the newer iPlanet Directory Server. (The Netscape Directory Server will no longer be used; its place is taken by the iPlanet Directory Server.) Tracy creates an instance of the Universal Connector (UTC) from Sun ONE Console which also adds a Connector View for the data on the newer iPlanet Directory Server by:

- Naming the instance: `firstgroup`
- Naming the view ID: `fg`
- Naming the base dn: `o=firstgroup`

  The LDIF file is already configured as `uid=variable,o=firstgroup` therefore, the base DN of our connector is input as `o=firstgroup`.
- Entering the LDAP URL of the Directory Server on which the data will be stored
- Entering an authenticating Distinguished Name and password

When Tracy refreshes the Directory Server and Meta-Directory Consoles, the *o=firstgroup* LDAP sub-tree is visible in the navigation tree of the Directory Server and the `firstgroup` connector and Connector View are visible in the navigation tree of Meta-Directory.

Once the Connector View is created, Tracy imports the LDIF file into the Connector View sub-tree via the iPlanet Directory Server by using the `Import Database` command from the Directory Server Console.

Flowing LDAP Data to the Meta View

The LDIF data is now written to a LDAP Directory Server that is directly connected to the Join Engine. Therefore, Tracy can begin configuring the join process to flow the LDAP data through the Join Engine and into the Meta View.by creating the rules to define how Siroe would like the data to flow.

Constructing an Attribute

destinationIndicator is an allowed attribute of the `organizationalPerson` object class so Siroe will use this attribute to keep track of an employee's off-site locations. Tracy, therefore, needs to put a value to this attribute. One way to do this is to take a value that is already present and break it into sub-tokens, enabling the use of these sub-tokens for other attributes. As their personnel's email address changes with each job, Tracy constructs an attribute with a value by:

- Naming the constructed attribute: `constructeddestinationIndicator`
- Entering token assignments: `{mail}=%mailto%@%company%.%com%`
- Entering attribute construction: `%company%`

With this rule, Tracy deconstructs the email address field, *variable@company.com* into three tokens: `variable`, `company` and `com`. From these fields, `%company%` is chosen as the value for `constructeddestinationIndicator` that will sync with the LDAP *destinationIndicator* attribute indicating the employee's off-site location.

Mapping Attribute Flow Rules

After constructing the attribute, Tracy configures flow tables to match up the attributes from the LDIF/LDAP data to the attributes in Meta-Directory's schema. Attribute flow rules can be configured for two directions: from the Connector View to the Meta View and from the Meta View back to the Connector View. The LDIF/LDAP data contains the following attributes: `objectclass`, `cn`, `sn`, `mail`, `title`, `uid`, and `givenname`; in addition, Tracy needs to flow the constructed attribute `constructeddestinationIndicator`. Tracy creates attribute flow rules by:

- Naming the rule: `fg2mv`
- Choosing the direction: `To Meta View`
- Entering selection criteria: `(%objectclass%==person)`

  This specification means that any entry with an `%objectclass%==person` attribute/value pair will be selected and allowed to map its attributes.

- Adding attribute mappings

  objectclass, cn, sn, mail, title, uid, givenname and constructeddestinationIndicator map respectively to objectclass, cn, sn, mail, title, uid, givenname and destinationIndicator

All attributes from the LDIF/LDAP data will flow to the same-titled attribute in the Meta View (although Tracy could flow them differently if Siroe wanted different mappings). The constructed attributes are visible when Constructed Attributes is chosen from the source objectclass drop-down list. In this case, Tracy maps the constructeddestinationIndicator constructed attribute to the schema attribute destinationIndicator.

When the attribute mappings are complete, Tracy needs to create a rule set entitled affg2mvset and add the fg2mv rule to it. Since attribute flow rule sets can not contain rules configured for data that flows in opposite directions, Tracy must also make a separate rule and rule set to map the attributes from the Meta View back to the Connector View. He names them fg2cv and affg2cvset, respectively. (The difference between the two rules is that, in the latter case, Tracy does not want the constructed attribute to be flowed back to the Connector View from the Meta View and therefore does not map it.)

Creating the Entry's Distinguished Name

Tracy needs to configure the distinguished name (DN) for each entry. DN mapping rules create a place for each entry in the destination view. The first entry in the LDIF/LDAP data has as its source DN uid=aquinn,o=firstgroup. Tracy wants the Meta View DN of this entry to be uid=aquinn,ou=Writers,o=firstgroup. The DN mapping rule is created by:

- Naming the rule: fgwriter2mv
- Entering the selection criteria: ((%objectclass%==person) AND (%title%==Writer))

  This specification means that any entry with %objectclass%==person and %title%==Writer attribute/value pairs will be selected and assigned the following DN.
- Constructing the DN: uid=%uid%,ou=Writers

  Tracy wants the unique identifier (in this case, aquinn) to be the unique name of each partial DN relative to the base DN. This DN construction creates the organization unit *Writers* in which all entries with title=Writer and objectclass=person will be placed.

Tracy must now configure a rule for Accountants and Designers. The three completed rules, `fgaccountant2mv`, `fgdesigner2mv` and `fgwriter2mv`, are then added to one rule set that Tracy calls `DNrulesfg2mv` because they are all for flowing data to the Meta View.

Tracy then makes three more rules which flow the information back to the Connector View, using as selection criteria `cn` in `ou=Writers, ou=Accountants`, and `ou=Designers` and constructing a DN of `uid=%uid%, o=firstgroup`. Tracy puts them in a rule set entitled `DNrulesfg2cv`. The two rule sets `DNrulesfg2mv` and `DNrulesfg2cv` will flow the LDAP data into and out of the Meta View. Next, Tracy flows the CSV data into the Meta View.

Mapping CSV Data to LDAP Using the UTP

The CSV data must be flowed into the Connector View using the Universal Text Parser (UTP), so Tracy configures an instance of the Universal Connector (UTC) from Sun ONE Console by:

- Naming the instance: `secondgroup`
- Naming the view ID: `sg`
- Naming the base dn: `o=secondgroup`
- Entering the LDAP URL of the Directory Server on which the data will be stored
- Entering an authenticating Distinguished Name and password
- Entering the path name to the `template.pl` script included with Meta-Directory: *NETSITE_ROOT*`/md50/utc-secondgroup/utp/template.pl`

  The `template.pl` file is part of the UTP. It is a Perl script that tells the UTC what to do. It must be moved to the configured UTC's directory (entered in the path name above) from its default location at *NETSITE_ROOT*`/bin/utc50/install/templates/universalparser/template.pl`.

Tracy refreshes both Console so that the *o=secondgroup* sub-tree is visible in the navigation tree of the Directory Server and the `secondgroup` Connector View is visible in the navigation tree of Meta-Directory. Now, Tracy needs to convert the data in the CSV file into LDAP data using the Universal Text Parser's `csv.cfg` file.

NOTE When using the Universal Text Parser, attributes are mapped twice. First, CSV data is mapped to LDAP attributes within the csv.cfg file and second, the CSV/LDAP attributes are mapped to the Meta-Directory schema by configuring indirect connector rules.

Mapping CSV Data to LDAP Attributes

The csv.cfg text file is used to map comma-separated values to LDAP attributes. Tracy finds the file and the task.cfg file in NETSITE_ROOT/bin/utc50/install/templates/universalparser. Tracy first copies the csv.cfg file to the task.cfg file. (task.cfg is called by template.pl but it needs to be configured specifically for CSV conversion into LDIF using the csv.cfg file.) Tracy edits the new version of task.cfg by editing the LineFormat and ImportLineFormat sections to map the CSV data attributes REALNAME, LASTNAME, EMAIL, PHONE, BUILDING, JOBTITLE, ALTERNATIVENAME and NICKNAME to the LDAP attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName, respectively. Tracy also changes the InputFile section of csv.cfg to %ScriptBase%csv.data. Tracy then places the csv.data file and the modified task.cfg file in the directory with template.pl located at NETSITE_ROOT/md50/utc-secondgroup/utp/.

Mapping LDAP Attributes to the Connector View

Once the CSV data has been mapped to LDAP-recognized attributes within the csv.cfg file, these LDAP attributes need to be mapped to the LDAP attributes in the Connector View using indirect connector rules. First, Tracy specifies the attributes defined in csv.cfg as attributes that are available to the connector. For this, each attribute name is entered on the connector’s External Available Attributes panel. Once defined, the attributes are accessible for mapping data to and from the Connector View attributes. Tracy can now create attribute flow rules for the indirect connector by:

- Naming the rule: secondgroupflow
- Choosing the mapping type: Mappings for Locally Owned Objects

  This choice identifies the rule as applying to data flow from the external source to the Connector View.
- Match the external attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName To Connector View attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName

- Match the external attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName From Connector View attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName
- Choosing the mapping type: Mappings for Connector View Owned Objects

  This choice identifies the rule as applying to data flow from the Connector View to the external data source.
- Match the external attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName To Connector View attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName
- Match the external attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName From Connector View attributes cn, sn, mail, telephoneNumber, l, title, uid and givenName

NOTE Again, in order to flow data in both directions, the attributes need to be mapped in both directions AND under each mapping type.

Tracy can now apply the indirect connector rule by selecting the utc-secondgroup indirect connector and its corresponding secondgroupflow rule.

Flowing LDAP-configured CSV Data to the Meta View

From this point, Tracy follows the same procedure established for the LDIF/LDAP data. First, DN mapping rules are configured, selecting the entries and linking them to the DN as laid out in the table below.

Table 3-2 Entries and their corresponding DN constructed rules

Select entries that contain	Link entries to DN constructed as
objectclass=person AND l=15	uid=%uid%, ou=Accountants
objectclass=person AND l=7	uid=%uid%, ou=Writers
objectclass=person AND l=18	uid=%uid%, ou=Designers

Tracy names the rules `sgwriters2mv`, `sgdesigners2mv` and `sgaccountants2mv` and puts them in a rule set called `DNrulessg2mvset`. Whereas the LDIF file had employees identified by job title, the CSV data has them identified by building location with each building occupied by accountants, writers, or designers exclusively. These DN mapping rules place people in the correct sub-trees based on their building location.

Mapping LDAP-configured CSV Data to the Meta View

Attribute flow rules are needed to map the CSV/LDAP attributes in the Connector View to the LDAP attributes in the Join Engine's Meta View. Tracy creates attribute flow rules by:

- Naming the rule: `sg2mv`
- Choosing the direction: `To Meta View`
- Entering the selection criteria: `(%objectclass%==person)`

  This specification means that any entry with an `%objectclass%==person` attribute/value pair will be selected and assigned the following mappings.
- Adding attribute mappings

  `uid`, `objectclass`, `telephoneNumber`, `sn`, `mail`, `title`, `givenName` and `l` map respectively to `uid`, `objectclass`, `telephonenumber`, `sn`, `mail`, `title`, `givenname` and `l`

Tracy also creates a separate rule to map the attributes from the Meta View back to the Connector View; this is called `sg2cv`. Rule sets are created (`afsg2mvset` and `afsg2cvset`)to hold the two configured rules. With that done, all the rules needed to manage the flow of data from the CSV file are complete.

Flowing Oracle Data to the Meta View

Tracy must now flow the Oracle data to the Meta View. This data includes salary information and employee identification numbers. To make the flow possible, Tracy needs to configure the Oracle database as a Data Server that can be recognized by the Meta-Directory's Join Engine.

NOTE In order to bring Oracle data into Meta-Directory, the OCI libraries, available from Oracle, must be installed on the same machine in which the Join Engine is installed.

Configuring the Oracle Database as a Recognizable Data Server

In order to configure the Oracle database as a recognizable Data Server, Tracy accesses the Data Server panel from Meta-Directory Console and completes the general information by:

- Entering a name for the data server
- Entering an alias for the data server

  This is the Oracle connect string.
- Entering a user name that does not already exist within the Oracle database

  This name will be used by Meta-Directory's Database Connector as the privileged user.
- Entering the path name to where the OCI libraries are installed
- Using the default NLS Language Environment Variable

Next, Tracy selects the tables and columns that will participate in the data flow by:

- Choosing the Oracle columns for syncing

  The Uninstrumented Columns are the full list of Oracle columns that Tracy will choose from. The Instrumented Columns are the ones Tracy has selected: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY and DEPARTMENT_ID.
- Choosing (or nominating) a primary key for Meta-Directory's use

  The primary key is unique and not null among all the entries. Tracy chooses EMPLOYEE_ID.

The name and location for the change log tables and triggers are also chosen. (Tuning and DCNS schedule use the default option.) When these selections are saved, a script is generated which, when applied to the database, enables change detection on the selected tables and columns. Tracy then runs the script on the command line via SQLplus (part of Oracle client software) and creates both the Connector View and participating view from Meta-Directory Console.

NOTE The Oracle table data can never be seen through the Meta-Directory Console. Only when the data is synced in the Meta View can it be viewed.

Constructing an Attribute for the Oracle Database

When flowing data from an Oracle database, the `objectclass` attribute must always be constructed and flowed with the data as Oracle does not use object classes. Tracy creates an `objectclass` attribute by:

- Naming the constructed attribute: `dbobjectclass`
- No selection criteria is necessary
- Entering attribute construction: `top;person;organizationalperson;inetorgperson`

Tracy also constructs a `commonname` attribute by:

- Naming the constructed attribute: `dbcommonname`
- Entering selection criteria: `(%cv.FIRST_NAME%=@)`

  This specification means that any entry with a Connector View value of FIRST_NAME will be selected.
- Entering attribute construction: `%cv.FIRST_NAME% %cv.LAST_NAME%`

  The `dbcommonname` will be made from the Connector View's FIRST_NAME and LAST_NAME attributes.

To complete the constructed attributes, Tracy puts each rule in a separate constructed attribute rule set entitled `dbobjectclass` and `dbcommonname`, respectively.

Creating Attribute Flow Rules for the Oracle Database

Tracy can now create the attribute flow rules to map the Oracle columns to their counterpart LDAP attributes. This is done by:

- Naming the rule: `oracle2mv`
- Choosing the direction: `To Meta View`
- Entering the selection criteria: `(%cv.EMPLOYEE_ID%=@)`

  This specification means that any entry with a Connector View value of EMPLOYEE_ID will be selected for these mappings.
- Adding attribute mappings

  `EMPLOYEE_ID`, `FIRST_NAME`, `LAST_NAME`, `SALARY`, `DEPARTMENT_ID`, `dbobjectclass` and `dbcommonname` map respectively to `uid`, `givenName`, `sn`, `description`, `departmentnumber`, `objectclass` and `cn`.

This attribute flow rule is then put in an attribute flow rule set entitled `oracle2mvset`. An attribute flow rule going back to the Oracle column is also created (using the selection criteria `%uid%=@`) and put in a rule set entitled `oracle2cvset`.

Creating DN Mapping Rules for the Oracle Database

Tracy can now configure rules that will create DNs for the Oracle data. Three rules are needed to flow the data into the Meta View's three organizational units. A DN rule for Writers is created by:

- Naming the rule: `oraclewriter2mv`
- Entering the selection criteria: `((%cv.EMPLOYEE_ID%=@) AND (%cv.DEPARTMENT_ID%==10))`

  This specification means that any entry with a Connector View value of EMPLOYEE_ID and a department ID of 10 will be selected for this DN.
- Constructing an DN: `%cv.EMPLOYEE_ID%, ou=Writers`

This DN rule creates the organizational unit *Writers* in which all entries with an employee ID and a department ID of 10 will be placed. Tracy must now configure a different rule for accountants and designers. The three completed rules that apply to data flowing into the Meta View are put into one rule set entitled `DNrulesoracle2mvset`.

Tracy then makes three more rules which flow the information back to the Connector View, using as selection criteria `%mv.EMPLOYEE_ID%=@` and constructing a DN of `%mv.EMPLOYEE_ID%`. Tracy puts them in a rule set entitled `DNrulesoracle2cv`. The two rule sets `DNrulesoracle2mv` and `DNrulesoracle2cv` will flow the LDAP data into and out of the Meta View.

Integration of the Data With Join Rules

With the bulk of the rules configured for the three data sources, Tracy is now able to flow the data into the Meta View. Join rules need to be configured to synchronize the data correctly as each data source has different information flowing to the same Meta View entry. Tracy needs to configure join rules that look at all source entries and link only those that have passed certain criteria. Tracy creates a join rule for the Oracle database by:

- Naming the rule: `oraclecv2mv`

- Entering the selection criteria: (%cv.EMPLOYEE_ID%=@)

  This specification means that any entry with a Connector View value of EMPLOYEE_ID will be selected for joining.

- Constructing the join filter: cn=%cv.FIRST_NAME% %cv.LAST_NAME%

Tracy puts the Oracle join rule into a join rule set called oraclejrcv2mvset. For the LDIF data source the join rule is configured by:

- Naming the rule: ldifcv2mv
- Entering the selection criteria: (%objectclass%==person)

  This specification means that any entry with an object class equal to person will be selected for joining.

- Constructing the join filter: cn=%cn%

Tracy creates a join rule set called ldifjrcv2mvset and the LDIF join rule is put into it. Tracy then realizes that the join rule created for LDIF data will also work for the CSV data source because they both have been converted into LDAP. Now Tracy is ready to apply the join process rules.

Applying Rules to the Participating View

Tracy applies the join process rules to the correct participating views. (The participating view allows the Connector View to 'participate' in the join process.) From Meta-Directory Console, a participating view for the firstgroup, secondgroup and oracle Connector Views is added. It is here that the join process rules are applied as follows:

Table 3-3 (Description of the join process rules and their corresponding rule set name

Participating View	Join Process Rule	Rule Set Name
firstgroup	Attribute Flow To Meta View	affg2mvset
	Attribute Flow To Connector View	affg2cvset
	DN Mapping To Meta View	DNrulesfg2mv
	DN Mapping To Connector View	DNrulesfg2cv
	Filters	none configured
	Join Rules To Meta View	ldifjrcv2mvset
secondgroup	Attribute Flow To Meta View	afsg2mvset

Table 3-3 (Description of the join process rules and their corresponding rule set name

Participating View	Join Process Rule	Rule Set Name
	Attribute Flow To Connector View	`afsg2cvset`
	DN Mapping	`DNrulessg2mvset`
	Filters	none configured
	Join Rules To Meta View	`ldifjrcv2mvset`
oracle	Attribute Flow To Meta View	`oracle2mvset`
	Attribute Flow To Connector View	`oracle2cvset`
	DN Mapping	`DNrulesoracle2mvset`
	Filters	none configured
	Join Rules To Meta View	`oraclejrcv2mvset`

Once the Join Engine is started, the connectors are enabled and the views are refreshed, Siroe's LDIF data, CSV data and Oracle data will flow to the Meta View forming one complete source of employee information.

At this point, Tracy checks the Meta-Directory error logs to see if any of the data has failed to flow or link. He does this by issuing the following `grep` command in the log directory:

```
grep -i fail *.log
```

Had any errors been found, Tracy could use the Fix-It Tool to manually link the orphaned entries. Fortunately for him no errors were found so Tracy steps outside for lunch.

Chapter 4

Console Interface

Most Sun ONE products use the Console graphical user interface for navigating and configuring Sun ONE software. The Console always takes the same form and layout. The product / Console name can be found in the top left hand corner. The *navigation tree* can be found in the left panel of the Console and the product features interface in the right panel. This panel will change depending on what is clicked in the navigation tree. This chapter describes the different Consoles and how to recognize and customize them. This chapter contains the following sections:

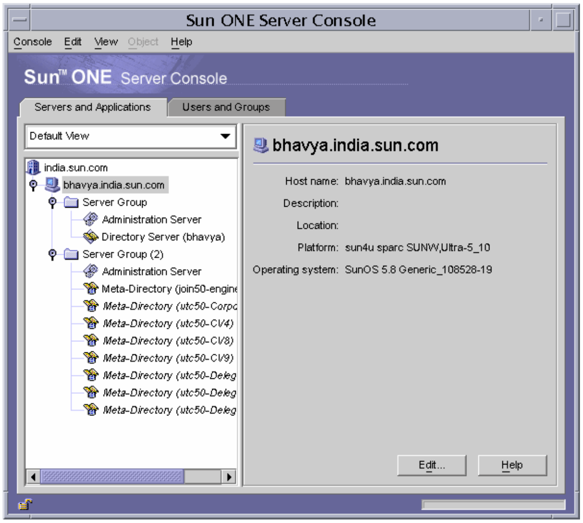
- Sun ONE Console Application
- Directory Console Application
- About the Console Window
- Menus

NOTE	The Consoles for all Sun ONE products are the Sun ONE Console application with integrated, product-specific JAR files.

Sun ONE Console Application

Sun ONE Console is the front-end Java application for the management of all Sun ONE software in your enterprise. It finds all the Sun ONE servers and applications registered in your configuration directory, displays them in a navigation tree, and allows you to manage and configure them. When you log in to Sun ONE Console, it connects to a server group’s Administration Server using the Hypertext Transfer Protocol (HTTP) although the two function independently. For more information, see the documentation for Sun ONE Console and Administration Server, Managing Servers With Sun ONE Console. The figure below shows Sun ONE Console's interface.

Figure 4-1 Sun ONE Console


Sun ONE Server Console
Console Edit View Object Help
Sun™ ONE Server Console
Servers and Applications
Users and Groups
Default View
india.sun.com
bhavya.india.sun.com
Server Group
Administration Server
Directory Server (bhavya)
Server Group (2)
Administration Server
Meta-Directory (join50-engine
Meta-Directory (utc50-Corpc
Meta-Directory (utc50-CV4)
Meta-Directory (utc50-CV8)
Meta-Directory (utc50-CV9)
Meta-Directory (utc50-Deleg
Meta-Directory (utc50-Deleg
Meta-Directory (utc50-Deleg
Edit...
Help

bhavya.india.sun.com

Host name: bhavya.india.sun.com

Description:

Location:

Platform: sun4u sparc SUNW,Ultra-5_10

Operating system: SunOS 5.8 Generic_108528-19

➤ To access a console window

1. Double-click the icon representing the Administration Server, Directory Server, or Meta-Directory from the navigation tree on the left of Sun ONE Console.

   The interface for that product will appear in the right panel.

2. Press Open in the upper right corner of the component information panel.

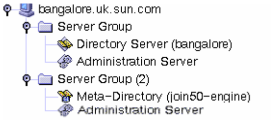
   This displays the Console window for that component in a new window.

Directory Console Application

By starting from within the Sun ONE Directory Server 5.2 installation, the Directory Server version of Sun ONE Console is displayed. The functionality of these two Consoles are the same. However, it is recommended to use the Console application appropriate to the software being administered.

About the Console Window

The Console window can be accessed by placing the mouse over the appropriate place in the tree appearing at the left side of the main Console. It is recommended that the Directory Server Console window be accessed from the Directory Server `startconsole` script (found within the Directory Server installation) and the Sun ONE Console windows be accessed from the Meta-Directory `startconsole` script (found within the Meta-Directory installation).

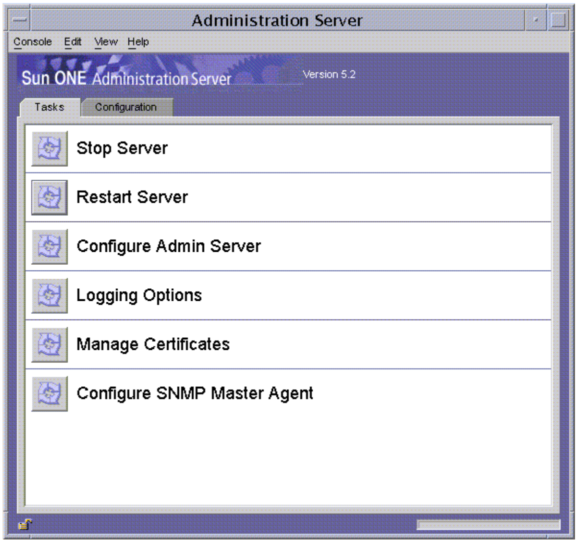


Sun ONE Administration Server Console Window

This has exactly the same functionality as the Directory Server Administration Server Console window mentioned in the previous section.

- Start the Sun ONE Console using the startconsole executable within the Meta-Directory installation
- Select, using the mouse, Server Group (2) > Administration Server

Figure 4-2 Sun ONE Administration window



Directory Server Console Window

You can perform most Directory Server administrative tasks from the Directory Server Console window. The Directory Server Console window offers navigation options as well as the task interface. For more information, see Sun ONE Directory Server documentation.

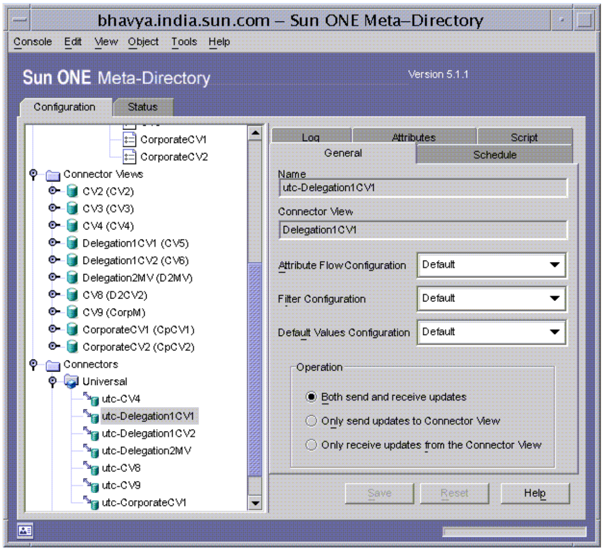
- Start the Directory Server Console using the `startconsole` executable within the Directory Server installation.
- Select Server Group > Directory Server.

Sun ONE Meta-Directory Console Window

The Sun ONE Meta-Directory Console window offers management, configuration, navigation and feature options for Sun ONE Meta-Directory software.The Sun ONE Meta-Directory Console window is pictured below.

- Start the Sun ONE Console using the startconsole executable within the Meta-Directory installation
- Select Server Group (2) > Meta-Directory. The following screen will appear:

Figure 4-3 Sun ONE Meta-Directory Console



Name: utc-Delegation1CV1

Connector View: Delegation1CV1

Attribute Flow Configuration: Default

Filter Configuration: Default

Default Values Configuration: Default

Operation

● Both send and receive updates

○ Only send updates to Connector View

○ Only receive updates from the Connector View

Menus

The Sun ONE Meta-Directory Console window has a number of features that are accessible from within the graphical user interface as well as from the menus at the top of the Console.

Console

From the Console menu, you have the following choices:

- Logon As A New User

  When provided with a new DN and password, your current session will end and the Console will re-open with another user as administrator.

- Close

  This option closes the Meta-Directory window.

- Exit

  This option ends your session with Meta-Directory.

Edit

From the Edit menu, you have the following choices:

- Preferences

  From the Console Preferences window, you can specify the directory where your settings will be stored as well as your choice of font for the Console window.

- Cut

  The standard command for use with Meta-Directory property editing tools.

- Copy

  The standard command for use with Meta-Directory property editing tools.

- Paste

  The standard command for use with Meta-Directory property editing tools.

- Delete

  The standard command for use with Meta-Directory property editing tools.

- Copy DN

  Choosing an entry from one of the Contents sub-trees in the navigation tree, selecting an entry in the right window and choosing Copy DN will copy the Distinguished Name of that entry to the Clipboard for pasting it into another field.

View

From the View menu, you have the following choices:

- Banner Bar

  The Banner Bar runs across the top of the Console and contains the Meta-Directory logo. Choose this option to show or hide the Banner Bar.

- Status Bar

  The Status Bar runs along the bottom of the Console and allows you to view the status of an operation. Choose this option to show or hide the Status Bar.

- Show Main Console

  Choose Show Main Console to bring the main Sun ONE Console interface forward.

- Tree

  Choose this option to show or hide the navigation tree on the left side of the Console.

- Refresh

  Choose this option to refresh the information in the navigation tree. When a new view is added, the Console should be refreshed to access the latest information.

NOTE The Refresh command always refreshes the entire navigation tree, even when you access the command by right-clicking on the components in the navigation tree.

Object

From the Object menu, you have the following choices:

- Open

  When the contents folder of a Connector View is selected in the navigation tree, the Open command opens up the Property Editor.

- Search

  When the contents folder of a Connector View is selected in the navigation tree, choose this option to search users and groups.

- New

  When the contents folder of a Meta View is selected in the navigation tree, choose New to create a new User, Group or Organization.

CAUTION The option of creating new users within the Connector View is available for testing purposes only; for instance, testing the flow between the external data source and its Connector View. Once the Join Engine and Meta View have been configured, new entries should only be added from within the Meta View or the external data source, not from within the Connector View.

- Start Server

  To start a Meta-Directory component, select it in the navigation tree and choose Start Server.

- Stop Server

  To stop a Meta-Directory component, select it in the navigation tree and choose Stop Server.

- Attribute Flow Summary

  Select Meta View in the navigation tree to see a summary of the attribute flow rules used by a specific view as described in the “Attribute Flow Summary” section of Chapter 1, “Configuring the Join Engine” in the *Administration Guide*.

Tools

From the Tools menu, you have the following option:

- Query

  The Query Tool is used to generate searches of Meta-Directory entries as discussed in “The Query Tool” section of Chapter 14, “Administration Tool” of the *Administration Guide*.

Help

From the Help menu, you have the following options:

- Contents

  This option links you to the contents window for the help available within the Meta-Directory.

- Index

  This option brings up a Search Index for the help files available within the Meta-Directory.

- Documentation Home

  This option brings you to the home page for Meta-Directory 5.1.1 Documentation.

- About

  This option brings up copyright and trademark information related to the Meta-Directory product.

Index