Integrating Service Manager® with Directory Services using LDAP

Best Practices for Integrating Directory Services with Service Manager using the Lightweight Directory Access Protocol (LDAP)

HP® Software Service Management

Introduction

As applications become increasingly distributed and reliant on networked computer systems, the need for communications among computers on the same local area network, within a corporate intranet, within extranets linking up partners and suppliers, or anywhere on the Internet, has increased as well. As such communications increase the complexity of administering distributed applications increases.

Information about the services, resources, and users that are accessible from applications must be organized clearly and consistently. This information, which must simultaneously be shared among applications and protected from unauthorized use, is usually stored in a database called an *information directory*. Maintaining and accessing this data consistently in a controlled manner enables consistent and seamless integrations within a distributed environment.

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that defines a method for accessing and updating this type of information. Storing data in a directory and sharing it among applications saves time and money by minimizing administrative effort and system resources. Figure 1 shows an example where a directory is used by various distributed applications.

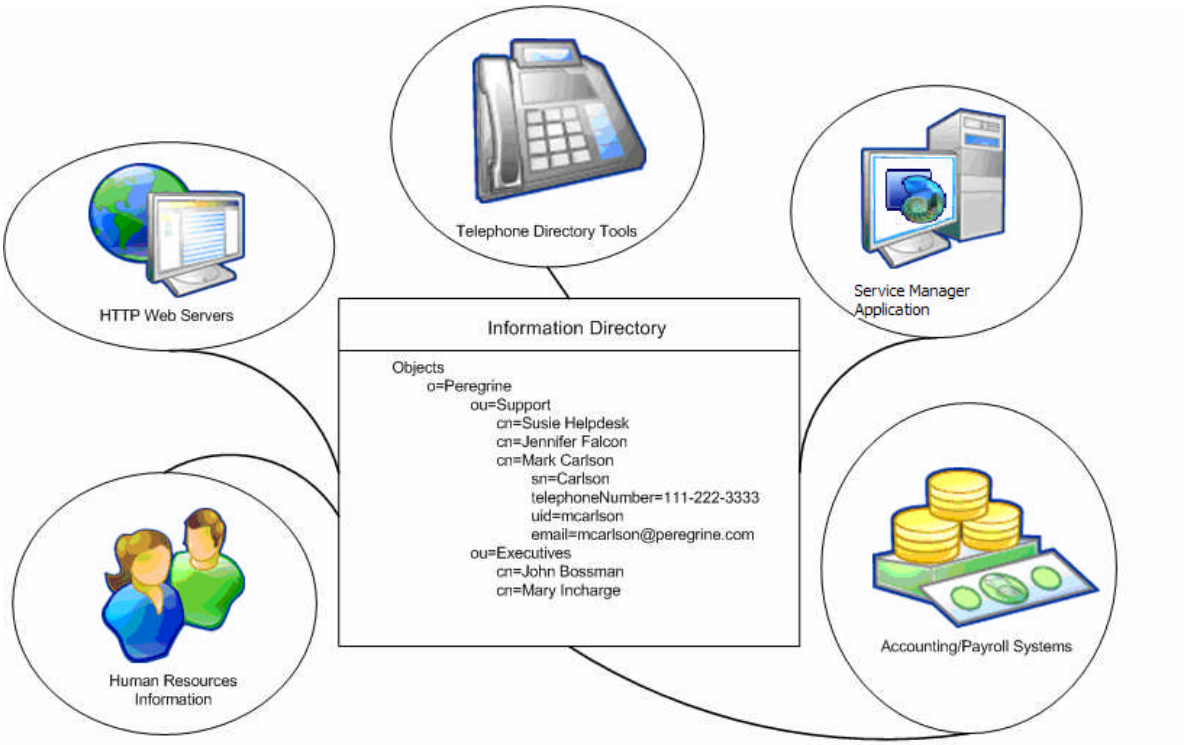


Figure 1: Information directory shared among distributed applications

The Service Manager LDAP interface is a server-side process that allows customers to use third party directory server data for Service Manager user authentication. This interface can also be used to implement standard data warehousing techniques that reduce the necessary amount of data storage and retrieval. The figure below depicts one such example.

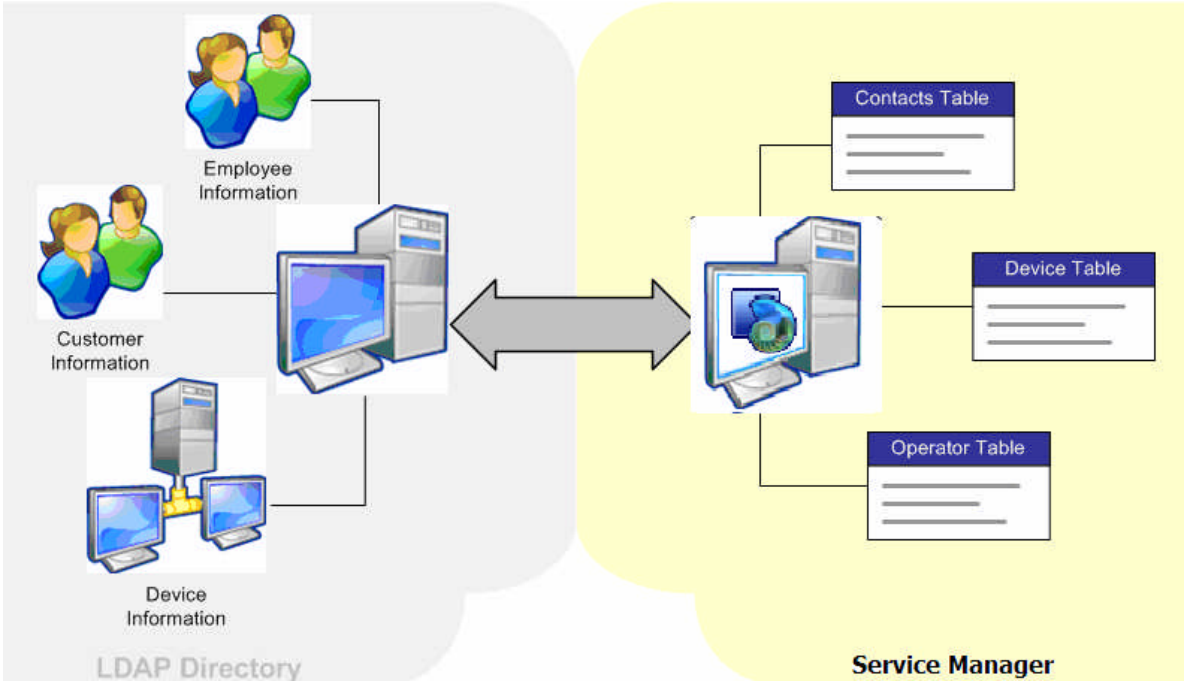


Figure 2: Using LDAP to Implement Data Warehousing Techniques

In this example, Service Manager retrieves information for the operators, contacts and device tables from the LDAP Directory Server. This information is not stored in Service Manager itself, which reduces the need for extra storage space and prevents user entry errors. All directory information can be accessed, retrieved, and controlled via the Service Manager LDAP interface.

Note: The Service Manager LDAP interface does not eliminate the need for a database that is associated with Service Manager. LDAP is not an alternative to a third-party RDBMS. LDAP simply allows certain information that Service Manager needs to be retrieved and updated in a common directory database.

Evolution of the Service Manager LDAP interface

Most enhancements that have been made to the Service Manager LDAP implementation were made in response to specific customer requests. During its initial release, LDAP was a relatively new interface and was not widely used by the business community. As the LDAP protocol and its use by Service Manager customers evolved, the Service Manager LDAP Interface adapted as well. The resulting interface uses various parameters as tailoring tools, rather than provide a generic interface that encompasses the basic functionality and intention of the LDAP protocol.

Figure 3 depicts the evolution of the Service Manager LDAP interface:

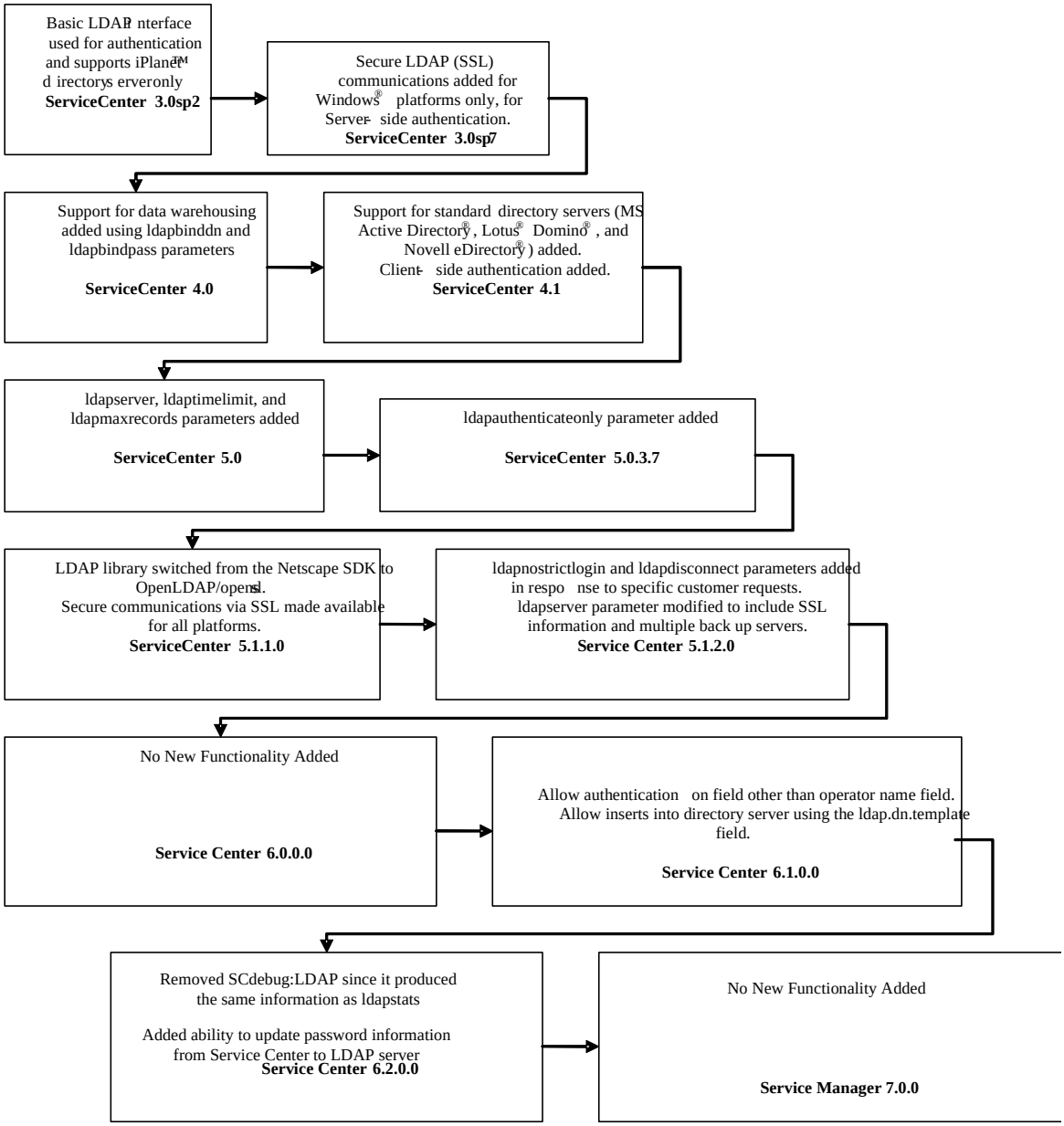


Figure 3: Evolution of the Service Manager LDAP interface

Planning your LDAP integration

Planning is an essential part of integrating Service Manager with a directory server. The following tasks are crucial to a successful implementation:

- Selecting a compatible directory server
- Defining your data warehousing needs
- Will the directory server be used (only) for authentication in Service Manager?
- Understanding your directory server configuration

Selecting a compatible directory server

There are numerous directory servers available for storing application information, and many claim to be LDAP v2 or LDAP v3 compliant. However, each has its own interpretation of the LDAP standard,

and anyone implementing Service Manager with LDAP needs to take that into consideration during integration. Some servers are more flexible and easier to integrate with than others.

For example, Sun™ ONE™ directory server allows anonymous access to its server for searching purposes. Microsoft® Active Directory® does not. This seemingly minor difference requires that special parameters be added to the Service Manager sm.ini file in order to connect to Active Directory. The differences in these interpretations of the LDAP standard are usually handled by Service Manager. The following directory servers comply with the LDAP versions mentioned above.

- Sun ONE directory server
- Microsoft Active Directory
- IBM® Lotus® Domino® Server
- Novell® eDirectory™
- OpenLDAP directory server

All samples in this document were written and tested against Sun ONE directory server version 6. If any of the samples provided do not work in any of the other LDAP directory servers, please contact customer support for assistance.

Defining your Data Warehousing needs

Storing data in a directory and sharing it among applications saves time and money by reducing administrative effort and system resources. This process is referred to as *data warehousing*. Any file in Service Manager can be mapped to a directory server so that it is updated and retrieved from that central location.

After a file has been mapped, updates are propagated between Service Manager and the directory server, but only if the user making the changes has the appropriate directory server rights. This is true for record insertion as well. New records added to Service Manager create new entries within the directory server if the user adding the record has the appropriate rights. Entries created in this manner contain values for only those attributes that are mapped to Service Manager fields.

Note: Deletions of records within Service Manager are propagated to the directory server if LDAP is set as the primary data source. Access rights can be modified on the directory server to prevent this. Make sure that data required by other applications is not removed from the directory server.

Part of planning the directory server interface is defining which Service Manager data you want to map, and determining the corresponding directory server attributes. Attribute names and data types often vary based on the directory schema. The figure below shows an example of contacts table mappings between a Novell eDirectory server and a Microsoft Active Directory server. Notice that the two servers use different attribute names to describe the same fields. The directory server administrator can provide the correct attribute names.

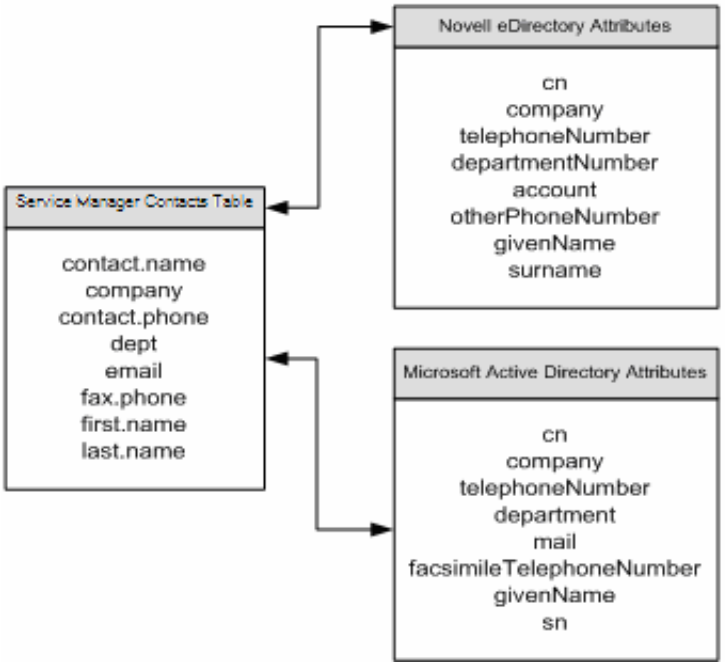


Figure 4: Example of table mappings that use different directory servers

Though authentication was the original intent of the Service Manager LDAP interface, it is not required. You can integrate any Service Manager table with a directory server for data warehousing without using LDAP for user authentication. User authentication and data warehousing are both possible, but independent of each other.

Because Service Manager uses the LDAP mapping for the `operator` file to bind to the directory server, any fields in the `operator` file that are mapped are retrieved from the directory server. If you want to authenticate using LDAP but do not wish to perform data warehousing on the `operator` file, map only the unique key of the `operator` file and use the ldapauthenticateonly parameter.

If you are not using LDAP for authentication, user authentication via the `operator` file need not be set up, but the ldapbinddn and ldapbindpass parameters must be added to the `sm.ini` file so that Service Manager can bind to the directory server. The format for these parameters is:

- ldapbinddn:<*fully qualified distinguished name for bind account*>
- ldapbindpass:<*directory server password for the ldapbinddn user*>

Note: For security purposes you can encrypt the values of these parameters in the sm.ini file. Please refer to the documentation on how to do this.

Successfully integrating Service Manager with a directory server requires that you have a clear understanding of your data objectives. You need to know the extent to which the directory server will be involved in your Service Management workflow, and then determine required access rights and data fields. Make sure you can answer the following questions prior to integration:

- Will I use Service Manager or the directory server as my primary data source?
- Will I need to retrieve information from multiple directory servers?
- What is my failover plan if the directory server is non-functional?
- Will the directory server be used only for authentication into Service Manager?

Will I use Service Manager or the directory server as my primary data source?

The method that Service Manager uses to merge data requires that you first determine the primary data source. Ideally, the primary data source should be the most valid source of data that is available. This is typically achieved by restricting update access to this data to ensure that the data remains consistent and correct.

When a query is executed, Service Manager first searches the primary data source for matching records, and then searches the secondary data source for records that match the result set of the primary data source. These data sets are then merged. Only records that exist in the primary source appear in the final set. The only exception is operator authentication with LDAP as primary data source. An operator template is used to merge data from both LDAP and Service Manager to ensure that all data needed by Service Manager, such as profiles, start menus etc. are set correctly.

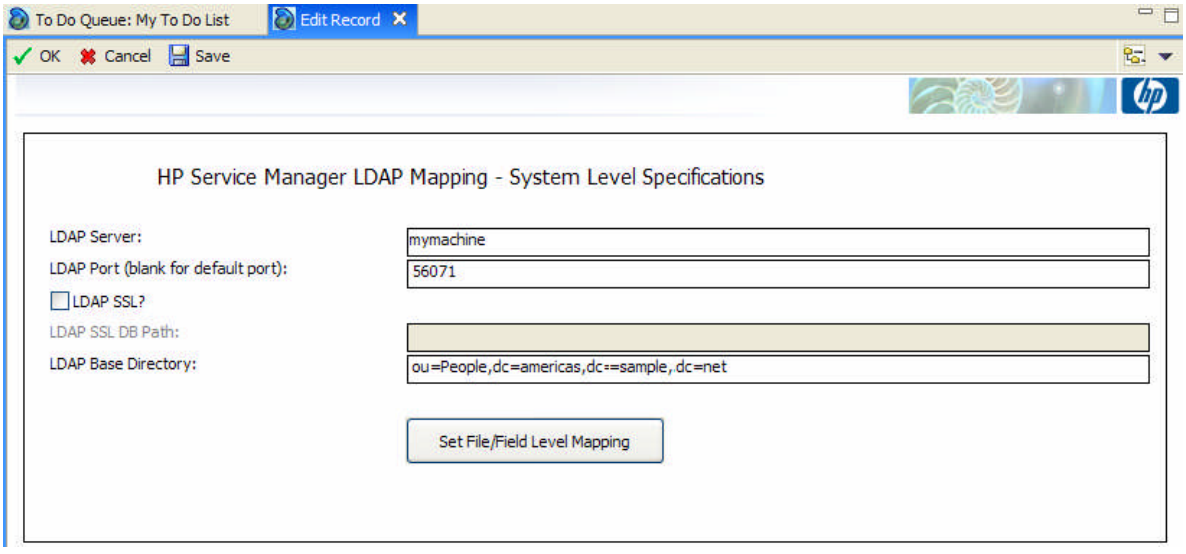
If the data in a mapped field differs between the directory server and Service Manager, the value from the primary data source (here: LDAP) overwrites that of the secondary data source (here: Service Manager).

The flexibility of the Service Manager LDAP interface allows the primary source to be set at the file level. For example, the contacts table can be mapped to use the directory server as its primary data source, while the device table is simultaneously mapped to the directory server, but uses Service Manager as its primary data source.

Note: If the directory server is set to be the primary data source, and the user executes a query that contains no LDAP mapped fields, Service Manager issues a query to LDAP for all its records. After merging these records with its local data, Service Manager performs the query's selection criteria to retrieve the appropriate data. If neither the directory server nor Service Manager has limitations on the number of records processed and on processing time, inaccurate results, including LDAP size or time limit errors, can occur.

Sample: Setting up LDAP as the primary data source

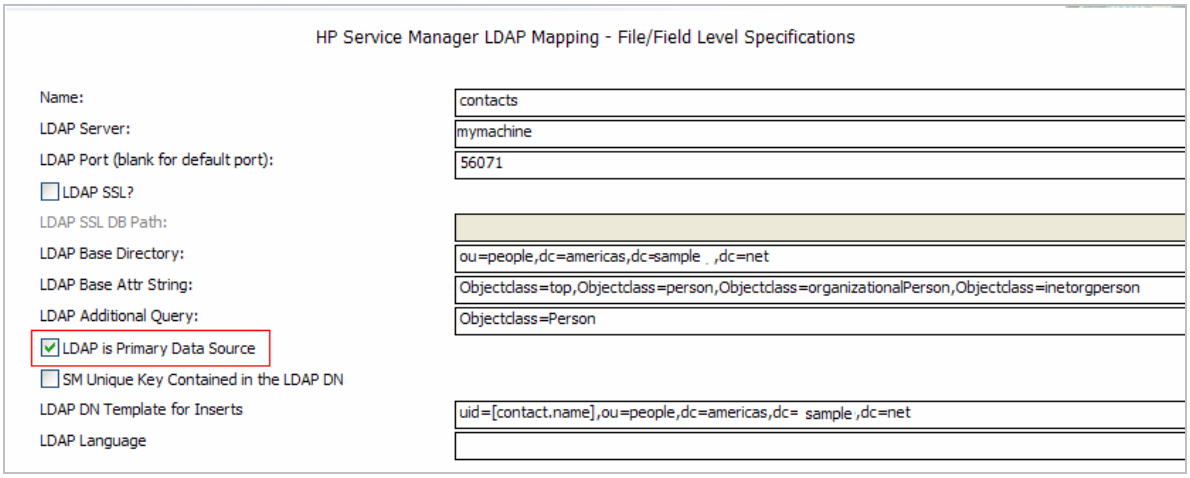
To set up LDAP as the primary data source, go to **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **System** – **LDAP Mapping**. Fill in all required connection information for the LDAP server:



Then click on **Set File / Field Mapping**.

Enter the name of the table to map to LDAP and click **Search**.

Check the **LDAP is Primary Data Source** checkbox as shown below:



LDAP Templates

When the `operator` file is mapped for authentication, and a user exists in LDAP but not in Service Manager, an operator record is permanently created in Service Manager for that user when he logs into Service Manager and is authenticated by the directory server. The authenticated user is granted pre-defined capabilities and is shown a login menu at startup. A complete operator record is created automatically by combining the mapped LDAP data with pre-defined Service Manager data, using an operator template and a default system record as follows:

- Service Manager attempts to create the user with information from both the operator template and the default system record.
- If neither the operator template nor the default system record exists, the operator record is not created and access to Service Manager is denied.

There are three different ways of assigning LDAP templates:

- *SYSDEFAULTS
- Template defined in the System Information Record (typically named the operator template)
- Individual / Role templates based on the systemplate field in the operator table

*SYSDEFAULTS is described in the next section.

After *SYSDEFAULTS is applied, the system looks for a template defined in the System Information Record. An operator template can be added there and stays in effect until changed or removed. An operator template is an operator record for first-time users and whose logins are authenticated by the directory server.

System administrators can design operator templates for creating operator records that share common information and settings. For example, you can design one operator template for managers and another template for first-level service desk operators. To assign the correct individual / role template to the operator, the systemplate field will have to be mapped to a field in the LDAP directory. If this field is not mapped, only the operator template can be used.

You can either create a new template from scratch or copy an existing record and modify it. You can design a template for any type of operator. Typically, this includes information such as:

- Application profile
- Default company
- Date information

- Time limits
- User session information
- LDAP information
- Security groups if Mandanten security is used

Any template record has to have the **Template Operator** checkbox checked in the operator table to prevent users from logging in without a password as these template users. An operator template record can not have a template that it uses itself. Any template entered for template operators is ignored.

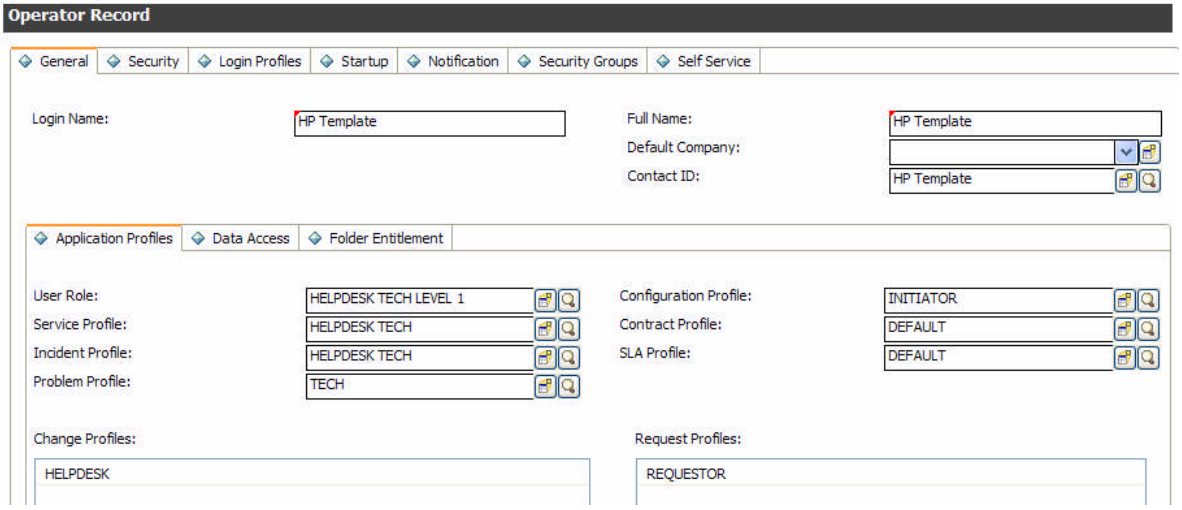
Note: See the Service Manager online help documentation for detailed instructions for creating and applying operator templates.

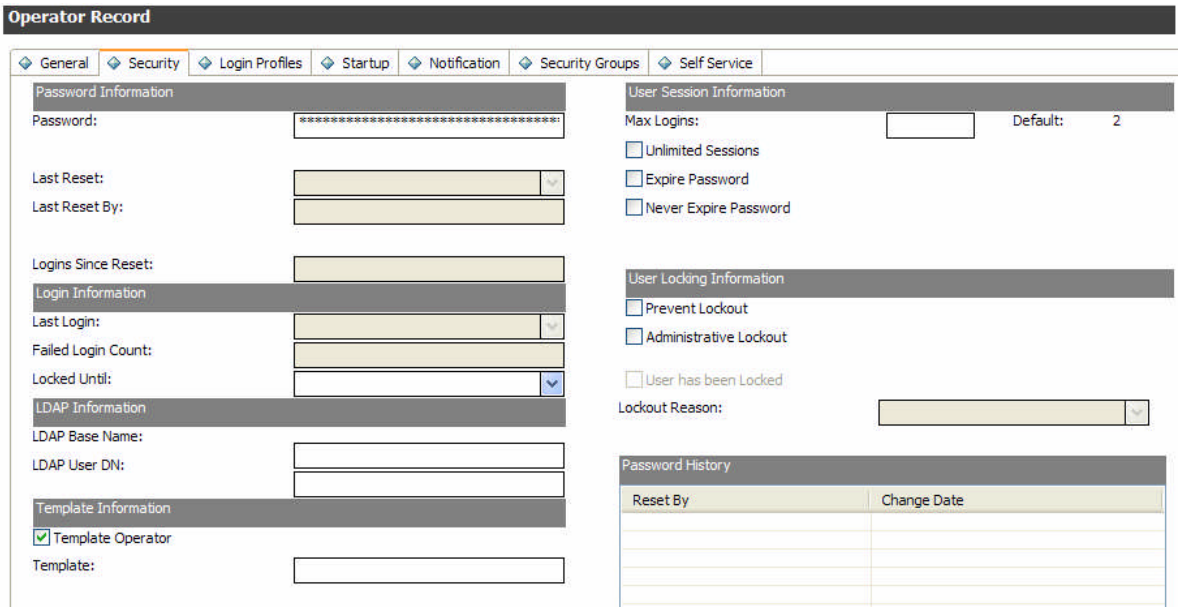
Sample template operator for LDAP

Note: To prevent the message from LDAP: *Message from LDAP server: Object class violation (se.base.method,add.record.radd)*, perform the following steps with **ldapdisable:1** in the sm.ini file. Otherwise, you can ignore that message.

Follow these steps to create the template operator for LDAP:

- Create a contact record to go with the operator.
  - To do so, go to **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **Contacts** and create the contact record with minimal information (contact.name and full.name are required).
- Then create the operator record.
  - Go to **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **Operators**. Enter the name of the template, e.g. **HP Template** in the Login Name field. All required fields that are needed by Service Manager but not mapped from LDAP need to be entered in this template.
  - Fill in the previously created contact record. Then fill in the most appropriate User Role – this will fill in all required profiles as well as the capability words and startup menu.
  - Click **Add** to add the record.
  - Then on the **Security** Tab, check the **Template Operator** checkbox.
  - Click **Save** to save the record.



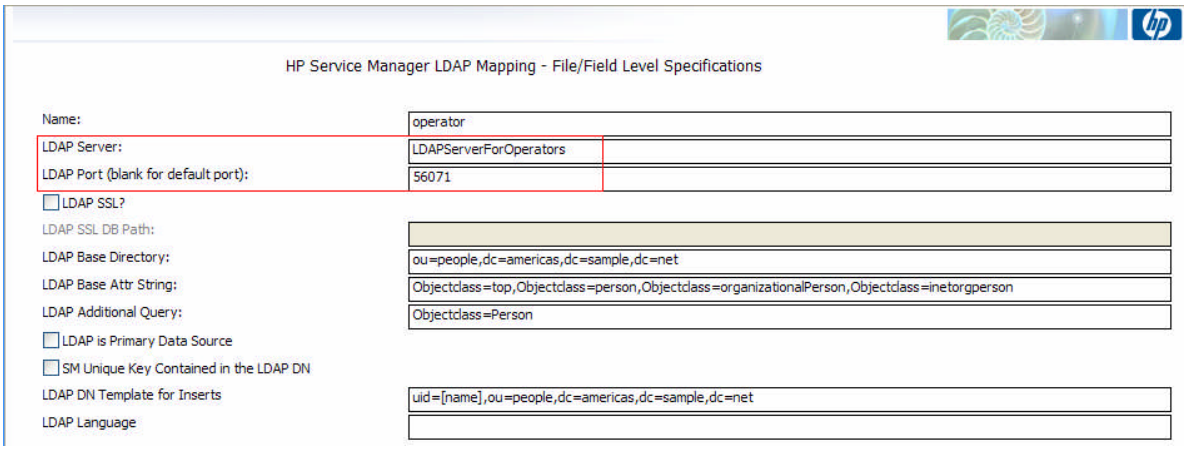


The Default System Record

A default system record, *SYSDEFAULTS, functions much like a template. It gets created by the System Administrator as a record in the LDAP-mapped table. It provides Service Manager with specific information about new users who are authenticated by a directory server. When a new user attempts to log in to a system in which the *SYSDEFAULTS record has been defined, a permanent operator record is created using both directory server data and values from the *SYSDEFAULTS record. Once added to the Service Manager database, operator records created in this fashion are independent of *SYSDEFAULTS, and are not altered by updates to *SYSDEFAULTS.

Will I need to retrieve information from multiple directory servers?

Service Manager lets you integrate a different directory server for each of its tables in order to support data retrieval from separate sources. For example, a company may store its customer information in one directory server and its employee information in a different directory server. To integrate both sources with Service Manager, the customer directory server would be mapped to the contacts table and the employee directory server would be mapped to the operator table. This mapping is defined in the `scldapfile` file using the Service Manager LDAP Mapping Tool. You can access this file by going to **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **System** – **LDAP Mapping**, and then clicking on **Set File / Field Level Mapping**. If it exists, Service Manager uses the mapping on the table level to connect to the directory server and ignores the information set in the `scldapconfig` record. The `scldapconfig` record can still be used as a default configuration for other tables that have no connection data in their `scldapfile` record.



Because multiple directory servers cannot communicate on the same TCP/IP port, make sure that the correct ports are specified in the LDAP mapping, and that they are not set to their default values. Assuming that your directory servers are not configured to be identical, you also need to set the base directory and filter information for each file.

What is my failover plan if the directory server is non-functional?

Because directory servers can become inoperable, directory server administrators typically set up backup servers to prevent major outages. The Service Manager LDAP interface can be set up to work with these failover plans using the ldapserver parameter in the `sm.ini` file. The ldapserver parameter defines the backup directory servers to which Service Manager can connect should the primary directory server become unavailable. Multiple directory servers can be defined by adding a number to the end of the parameter. For example:

ldapserver1:ldapbackup2,1230
ldapserver2:ldapops

Note: The only required attribute for the ldapserver parameter is the hostname, the port number should be entered if not the default. The option will only work for non-SSL connections.

Important: Previous versions of this document mentioned other parameters for this option as well. These parameters are:

```
host name, port, base directory, certificate file, key file (not used in
Service Manager)
```

Unfortunately, the option does not recognize the fact that the dn contains commas, so it interprets the second section of the dn (relative dn) as the certificate file and the third as the key file etc.

```
ldapserver1:ldapmain,636,"uid=users,dc=acme,dc=com",c:\certs\sslcert.pem,
c:\certs\key.txt
ldapserver2:ldapb,640,"cn=Joe
User,dc=acme,dc=com",c:\certs\sslcert2.pem,c:\certs\key2.txt
```

SCR 40704 was opened to address this issue in future versions.

Will the directory server be used (only) for authentication in Service Manager?

One of the primary functions of the Service Manager LDAP interface is user authentication. The `operator` file in Service Manager can be mapped via LDAP so that users are validated by the directory server, bypassing standard Service Manager authentication. If the directory server should only be used for authentication, use the ldapauthenticateonly parameter:

- ldapauthenticateonly – When enabled, Service Manager connects to the directory server for user authentication only. After the user is authenticated, the directory server connection is cancelled and the directory server is no longer used by Service Manager. All LDAP directory mappings are ignored, and data is retrieved from the database that is associated with Service Manager.

It is possible that not all users that need to log in to Service Manager are defined in the LDAP server, for example falcon as the Service Manager System Administrator. To enable these users to log in while LDAP is used for authentication, use the ldapnostrictlogin parameter:

- ldapnostrictlogin - Enabling this parameter allows a user who has a valid operator record to log into Service Manager without the need for a valid directory server record. If disabled, users are required to have a valid directory server entry in order to log into Service Manager.

Note: Service Manager does not allow logins without a password when mapped to LDAP. Each operator needs to have a password value set.

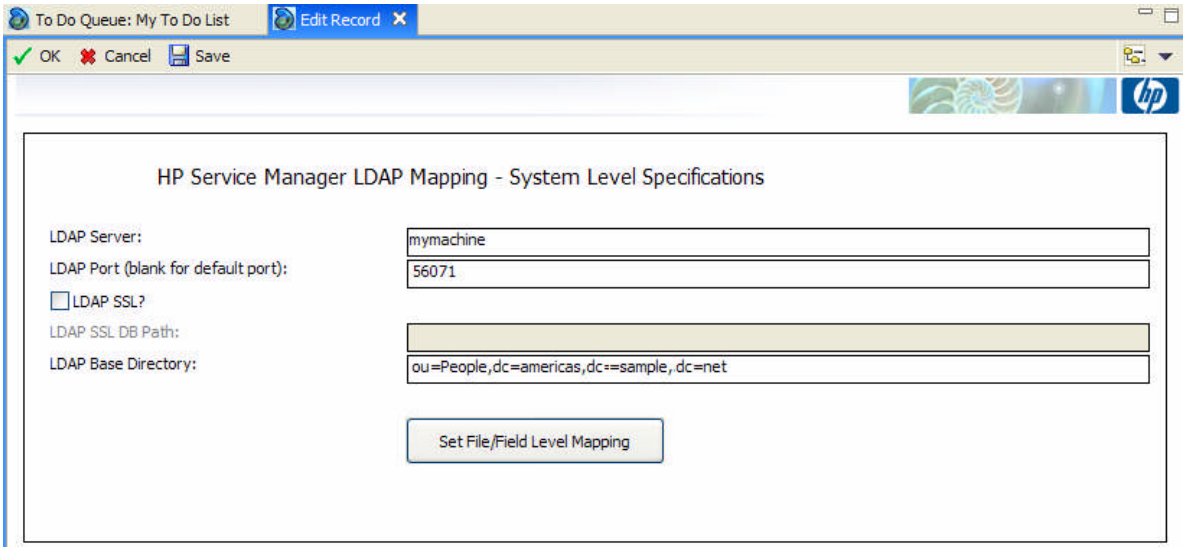
Sample Setup for authenticate only:

The following parameters should be added to the sm.ini:

```
ldapauthenticateonly:
ldapnostrictlogin:1
```

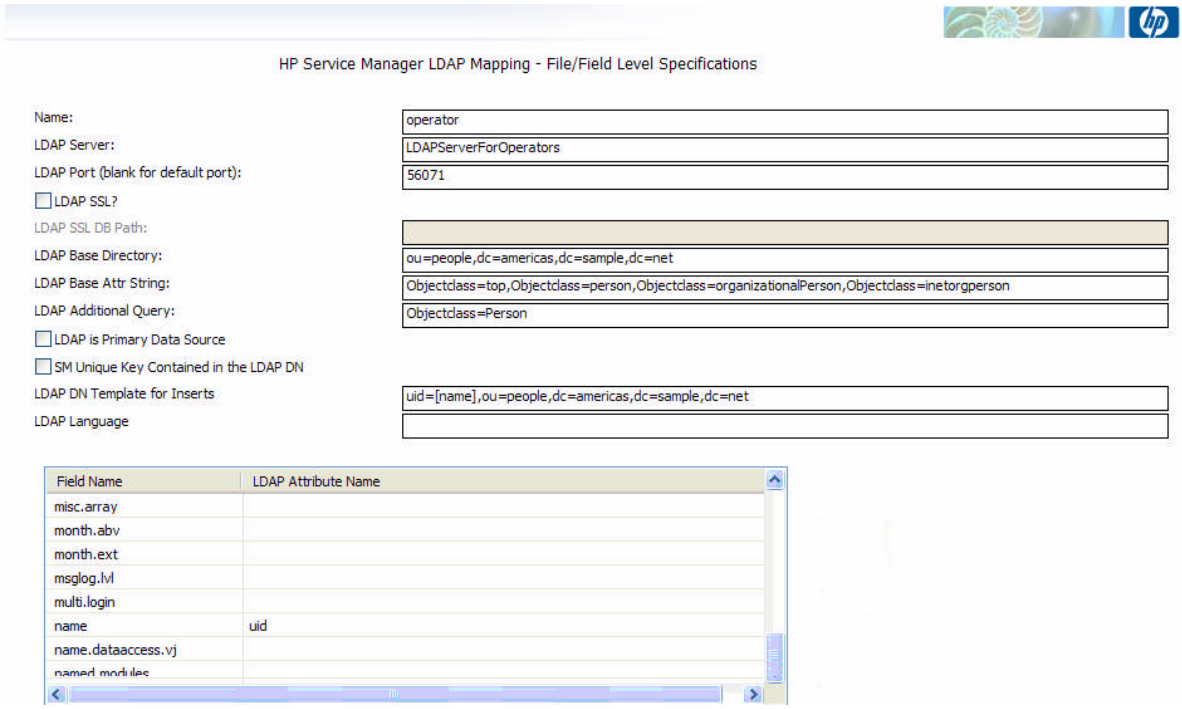
The LDAP settings in Service Manager should be as follows:

Go to **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **System** – **LDAP Mapping**:



Click on **Set File / Field Level Mapping**, then enter **operator** in the name field and click **Search** to map the appropriate fields of the operator table. For authentication purposes, only the operator **name** has to be mapped. Other fields may be mapped if needed for data warehousing in addition to authentication purposes.

Note: The password field only has to be mapped if the user should be able to update the password in LDAP via the Service Manager change password functions. Otherwise, do not map this field.



Understanding your directory server configuration

Directory servers are very user-specific, and can be configured to take advantage of numerous options that allow tailoring for specific requirements. The Service Manager LDAP interface is capable of handling most of these options out of the box. However, there are a few exceptions.

- Anonymous authentications
- Server referral chasing
- LDAP proxy servers with SSL

Anonymous authentications

In LDAP, authentication is accomplished using a bind operation. A client initiates a connection with the directory server by sending the server a bind operation that contains authentication information. A client that sends an LDAP request without sending a bind operation is treated as an anonymous client. Many directory servers do not enable this functionality out of the box, and LDAP administrators can disable it for tighter security. Anonymous binds are usually allowed when there is a need to provide access to a subset of the directory server's information to a third-party application that is not capable of authenticating to the directory server. The best way to determine whether your directory server allows anonymous operations is to contact your LDAP administrator.

If you determine that your directory server does not allow anonymous operations, add the ldapbinddn and ldapbindpass parameters to your `sm.ini` file. Service Manager uses the information from these parameters to authenticate to the directory server. The format for these parameters is as follows:

- ldapbinddn:*<fully qualified distinguished name for bind account>*
- ldapbindpass:*<directory server password for the ldapbinddn user>*

Note: For security purposes you can encrypt the values of these parameters in the sm.ini file. Please refer to the documentation on how to do this.

Sample:

```
ldapbinddn:cn=BOB.HELPDESK,ou=people,dc=americas,dc=sample,dc=net
```

```
ldapbindpass:password
```

Server Referral Chasing

A directory server can be configured to send a client a referral if the client queries for an item that includes a suffix that is not in the server's directory tree. The referral sent by the directory server indicates that the information that the client has requested can be found at another location, or possibly at another server. Referrals contain LDAP URLs that specify the host, port, and base DN of another directory server. A simple example of LDAP referral processing is depicted in Figure 5.

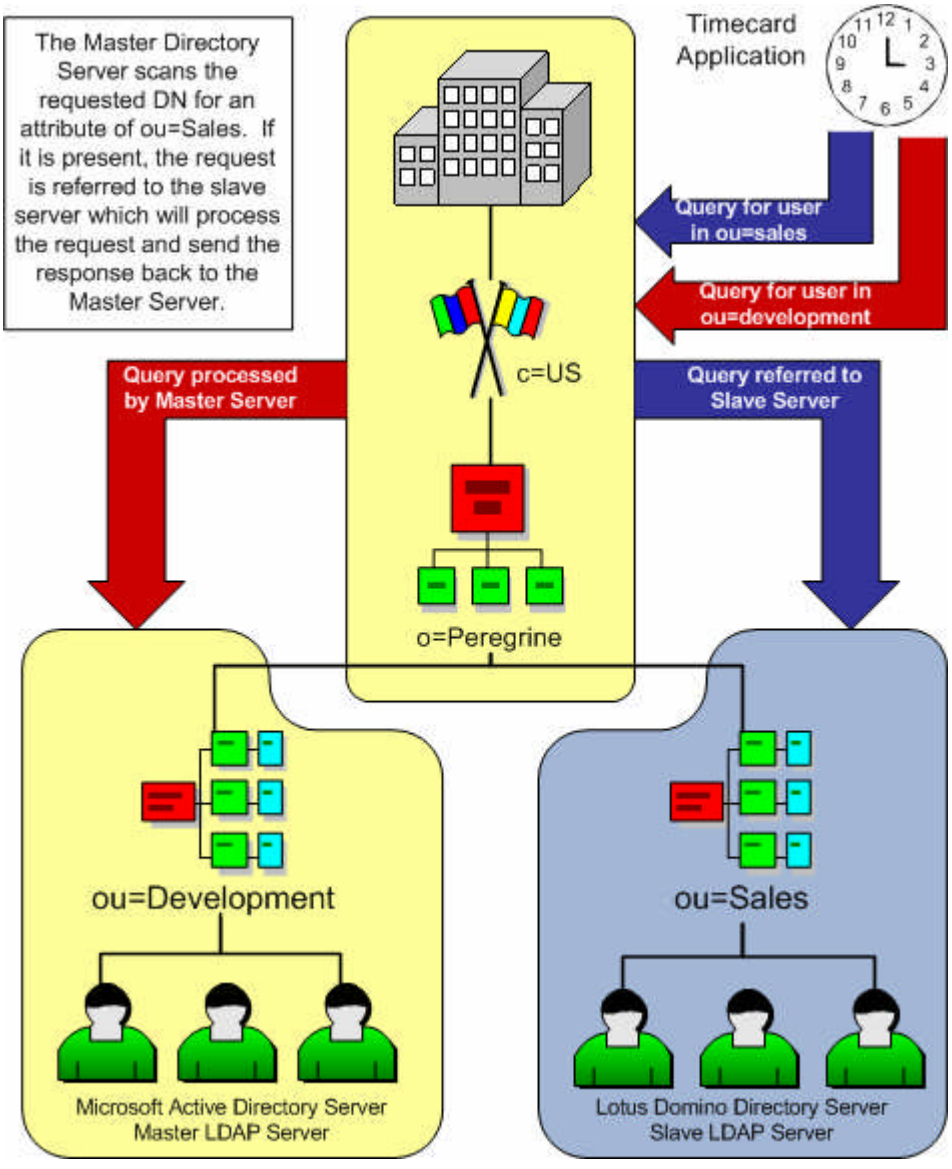


Figure 5: LDAP Referral Chasing

The company in Figure 5 has a development department that uses Microsoft Active Directory and a sales department that uses the Lotus Domino directory server. A new software application has been installed to track employee hours company-wide. This application needs to retrieve authentication information for users from these two separate directories. The company accomplishes the requirement of a single authentication method by issuing referrals from the Active Directory server to the Domino server.

In most cases, referral chasing is transparent to the directory server client. It is the responsibility of the underlying LDAP API to attempt to bind to any referrals and add the search results to the result set. If the referral points to an object in a different domain, the underlying LDAP API should attempt to use the current credentials to bind to the target of the referral. Service Manager 7 supports non-anonymous referral chasing.

LDAP proxy servers and SSL

An LDAP proxy server is a mediator between an LDAP client and one or more LDAP-enabled resources. The proxy server's role is to direct and transform queries transparently to the LDAP servers, and then to filter the responses back to the client. Proxy servers are often used to access resources outside a firewall. However, an LDAP proxy server can also provide a method for controlling access to resources outside the actual domain via the LDAP protocol. For example, you can use a proxy server to join different domains in an intranet.

LDAP proxy servers can add access control, authenticate users, restrict access to resources, and rewrite requests using regular expressions. They can also map to and hide from other servers. This type of server is frequently used for load balancing and fault tolerance, and can also include a cache to store results of frequently requested queries.

Reasons for using an LDAP proxy server include:

- Create a consolidated view of your internal LDAP directory resources.
- Provide load-balanced and failover access to your directory resources.
- Manage secure access from internal users to external or partner LDAP directory resources.
- Manage secure access from external parties to internal LDAP directory resources.
- Manipulate or transform information being passed to and from an LDAP query according to programmed business logic.
- Consolidate LDAP queries through one server to avoid referrals that clients are using.

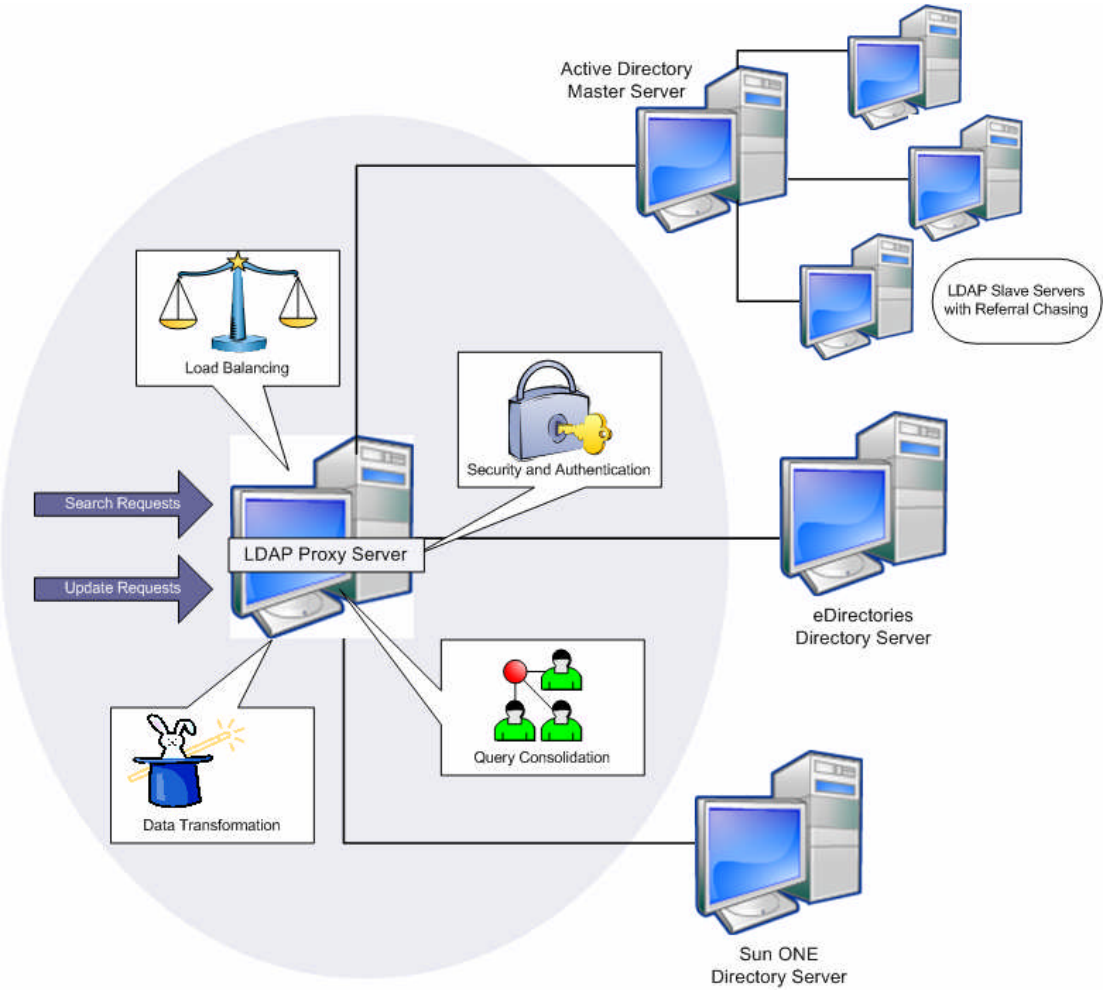


Figure 6: Typical uses for an LDAP proxy server

Even though an LDAP proxy server is transparent to Service Manager, give careful attention to the generation of your SSL certificates if your configuration uses SSL. Because the server name listed in the certificate must match the server name located in the DNS table, separate certificates need to be generated for the proxy server and its slave servers. This process is discussed in further detail in the section *Configuring the server-side SSL connection*.

Understanding the directory server schema

Once the directory server is in place, it is important to review its schema. A directory schema describes both the types of objects that a directory may include and the mandatory and optional attributes of each object type.

Note: Your directory server administrator should be able to provide details about the schema, and keep you informed about changes to it. The slightest change to the directory server configuration can affect your integration, and it is critical to know about such changes in advance.

Service Manager needs to know the following about the schema in order to integrate successfully:

- What is the basic format of the directory server?
- What is the format of the directory server's DN?
- What should I use as the base DN when searching the directory server?

What is the basic format of the directory server?

The LDAP API retrieves data from a directory sever by issuing search requests and processing the information that is returned. To maintain optimum performance from your directory server integration, searches should be as precise as possible. This not only decreases your search time, but also decreases the number of entries returned to the API. The key to keeping searches precise is to understand your directory server's *directory tree*, which represents the hierarchy of objects in the directory server. This hierarchy is derived from the schema and provides the basis for defining the scope of your search requests. It is essential to know which objects you are searching for and where they reside in the directory tree.

What is the format of the directory server's DN?

Every object in a directory server has a distinguished name (DN) that uniquely identifies it and its location within the directory tree. The DN is built using the object name and the names of the container objects and domains that contain it. Since the DN is unique for each object, it can be used to retrieve object- specific information from the directory. The following figure is an example of how a DN is derived.

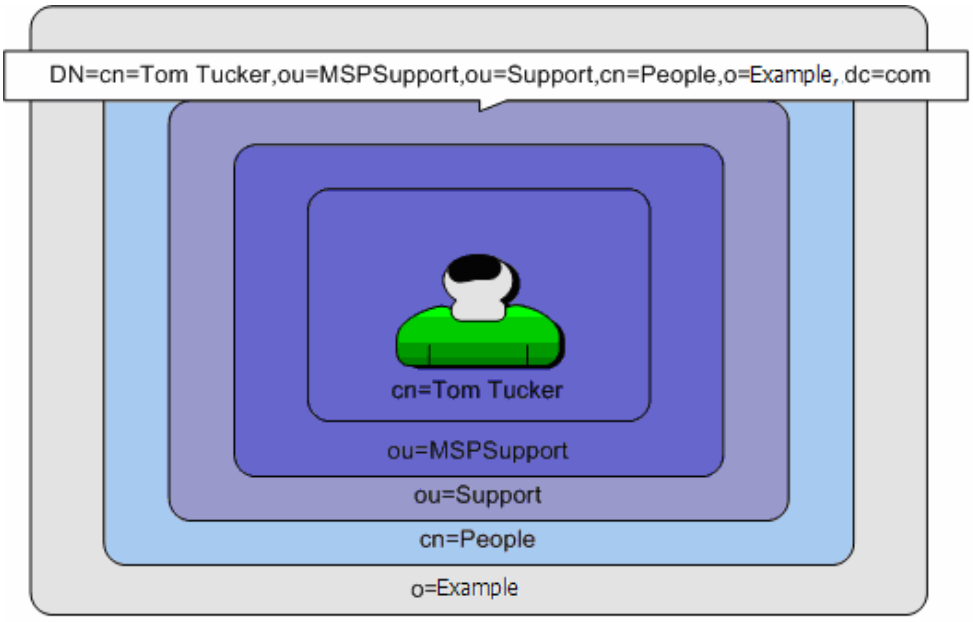


Figure 7: Formation of a distinguished name

Service Manager can bind to the directory server only with a valid DN. If you have specified the ldapbinddn and ldapbindpass parameters, Service Manager uses them as the binding DN and ignores the operator file settings. Otherwise during login, Service Manager searches for the user in the LDAP directory based on the operator name field, and retrieves the user's DN from LDAP. The SM Unique Key Contained in the LDAP DN flag and the LDAP Attribute Name of the field mapped to the primary key (name) are both used to create the DN.

The SM Unique Key Contained in the LDAP DN flag

Note: Setting this flag is not recommended (see Notes below) and will not work on Active Directory.

This flag is checked by Service Manager to determine whether to use the LDAP attribute name that is mapped to the primary key as the starting object for the DN.

This flag should be checked only if the starting object of your DN contains data that exactly matches the primary key of your Service Manager file. Figure 8 illustrates this concept with two separate examples that use the operator file.

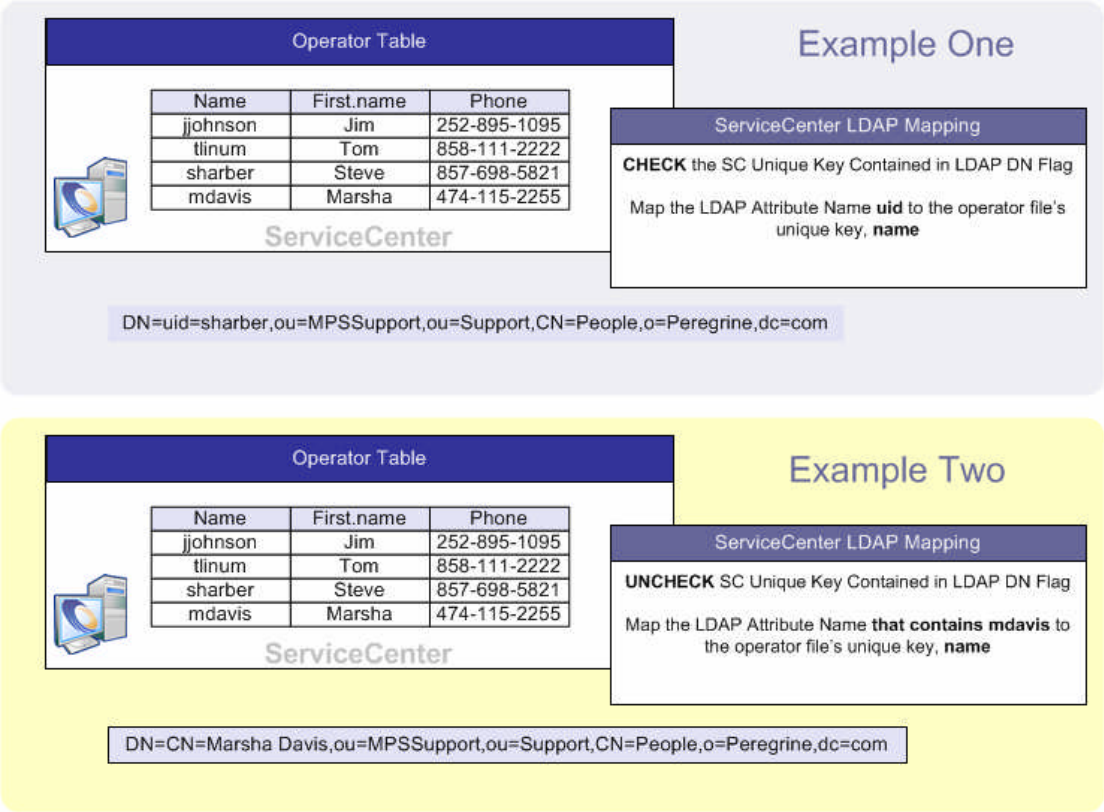


Figure 8: Use of the SM Unique Key Contained in LDAP DN flag

If the SM Unique Key Contained in LDAP DN flag is checked, Service Manager builds the DN by combining the LDAP Attribute Name field with the username entered at login and the LDAP base directory. Service Manager then attempts to issue a search for the DN; if it finds the DN, it tries to bind to the directory server with this DN and the password that was entered during login. If successful, the user is considered authenticated and logged into Service Manager. If unsuccessful, the user is denied access and receives an "Error 49 Invalid Credentials" message from LDAP.

Note: There is a known issue with the SM Unique Key Contained in LDAP DN flag. If this flag is set and the operator logging in is not located in the base directory, authentication fails. If any of your authenticating users are located in any of the sub-trees of your base directory, do not set this flag.

If the SM Unique Key Contained in the LDAP DN flag is not checked, Service Manager uses the ldapbinddn parameter, if set, to bind to the directory server. If the parameter is not set, Service Manager attempts an anonymous search on a combination of the LDAP Attribute name and the username entered during login. If the search is successful, the exact DN is returned from the directory server and Service Manager uses it as the bind account for the Service Manager client from then on. If it is not located, the user is denied access and receives an "Error 32 LDAP Object Not Found" message from LDAP.

Note: If your directory server does not allow anonymous searches and the SM Unique Key is not contained in the LDAP DN, the ldapbinddn and ldapbindpass parameters must be specified.

Limiting the result set

For best performance it is important to limit the return set from LDAP to only contain relevant records. There are several ways of accomplishing this:

- Setting the correct base DN
- The ldapsearchscope parameter
- The LDAP Additional Query field
- The ldaptimelimit parameter
- The ldapmaxrecords parameter

Setting the correct base DN

Directory searching polices can vary, and it is important to understand the rules that your directory server follows when determining which base DN to use for searching. For example, some directory servers allow a search to start only at the bound user's level and work down to its children. In this case, you would set your bound user to be an account that is located at the highest desired search level. This practice could also be used to limit the entries that a user can access in the directory server. Figure 9 depicts an example of how you could separate customer support accounts from executive accounts, and control access to the different subsets of information.

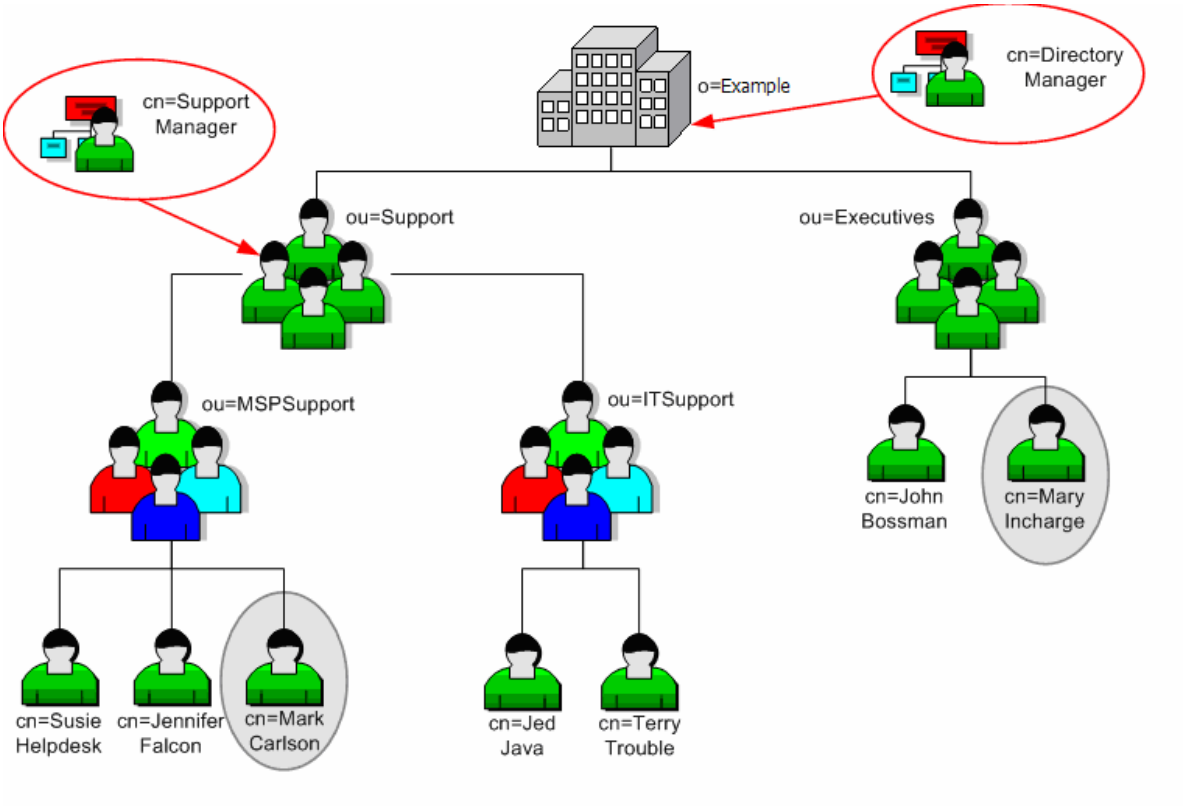


Figure 9: Choosing the appropriate bind account

In this example, Directory Manager and Support Manager are accounts on the directory server that have access to varying subsets of data. Because the Directory Manager account is located at the root level (cn=Directory Manager, o=Example), a connection bound by this user would have read access to all entries in the directory server. The user would be able to view both Mary Incharge located in the Executive organizational unit (cn=Mary Incharge, ou=Executives, o=Example) and Mark Carlson located in the MSPSupport organizational unit (cn=Mark Carlson, ou=MSPSupport, ou=Support, o=Example). Because the Support Manager account is located in the Support organization unit

(cn=Support Manager, ou=Support,o=Example) , this user would have read access to the MSPSupport and ITSupport organizational units, and would not be able to access Mary Incharge in the Executive organization unit.

In summary, keep your searches as narrowly focused as possible without restricting access to the information you need. The base DN is specified in the LDAP Base Directory field of either scldapconfig or scldapfile. Setting your base DN to a value that is too broad causes performance problems and presents the possibility of retrieving too much data. Setting it to a value that is too restrictive could keep objects from being located. An LDAP administrator who knows how the directory server is configured and how it handles searches can be a very useful resource.

The ldapsearchscope parameter

Service Manager can adjust the scope of a directory server search using the ldapsearchscope parameter. When ldapsearchscope is set to 0, all directory server searches begin at the LDAP base directory specified in the LDAP configuration record and continue through its sub-trees. If set to 1, only the LDAP base directory is searched.

The ldapsearchscope parameter is often used when a directory server contains a vast amount of dissimilar entries over numerous containers and organizational units, in an arrangement that is similar to the structure depicted in Figure 10.

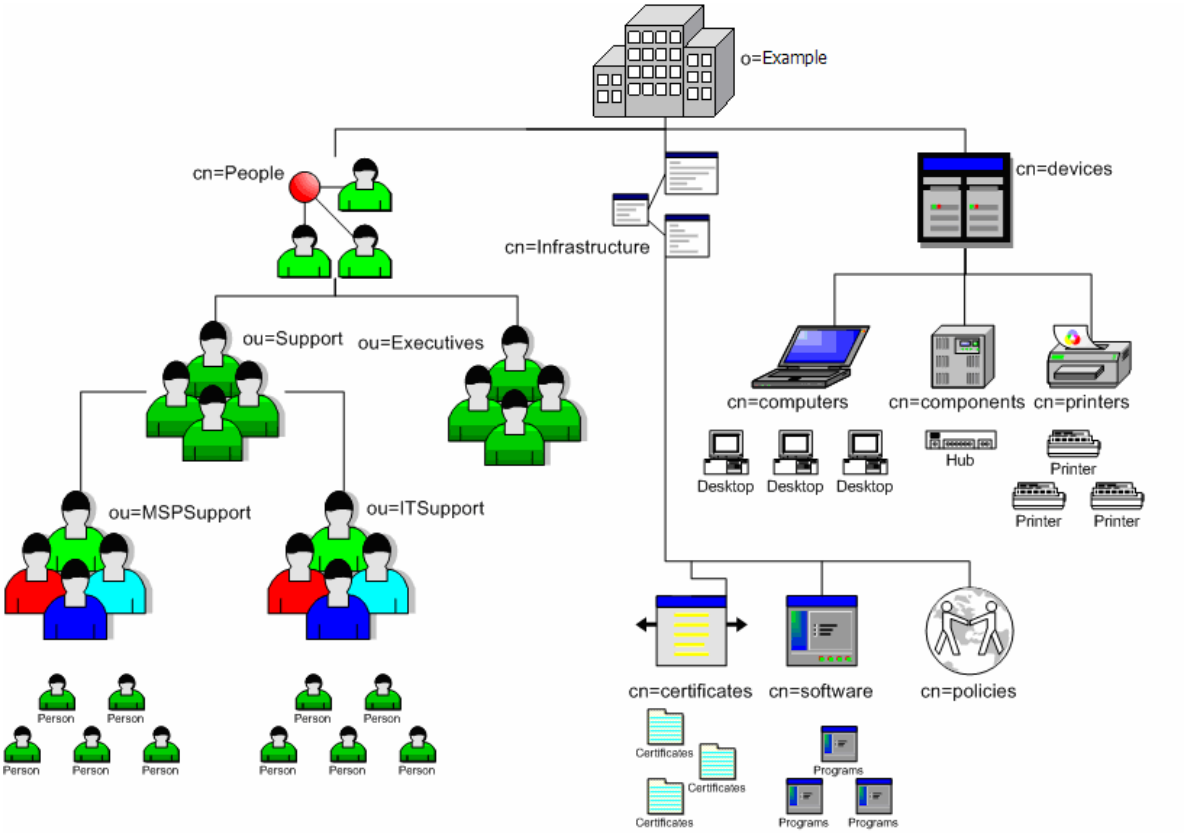


Figure 10: Complexity of searching a directory server

In this example, if the LDAP interface were set to search the entire base directory of o=Example, a query for a person located in the ITSupport organizational unit would also search through the devices and infrastructure containers, which could contain numerous entries. This would make the query time consuming and extremely inefficient. If we knew, for instance, that the query would always be for a member of the ITSupport organizational unit, we could set the base directory to ou=ITSupport, ou=Support, cn=People, dc=Example, and the ldapsearchscope parameter to 1. This would limit the search to the ITSupport area only and greatly increase the query's performance.

The LDAP Additional Query field

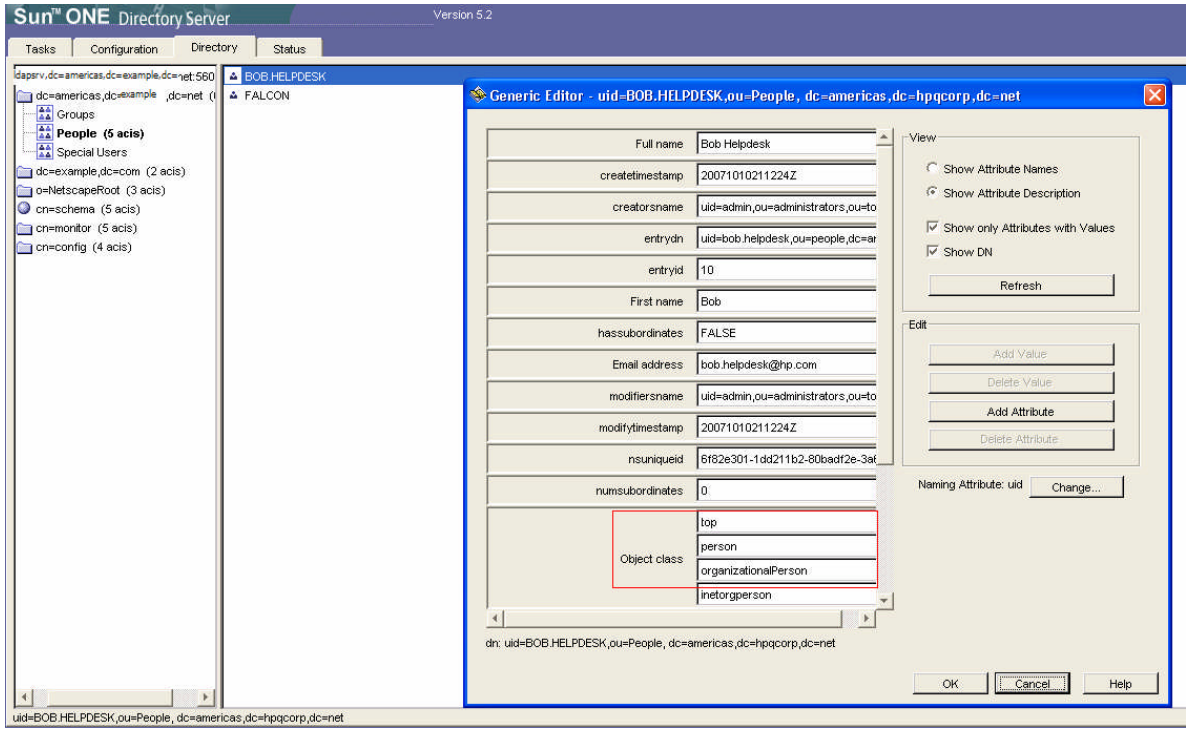
The Service Manager LDAP Additional Query field allows users to define a search filter that will be attached to LDAP query Service Manager submissions. Search filters allow users to achieve more effective and efficient searches. The value in this field should be a valid search filter string as specified in the LDAP Standard RFC 2254, "The String Representation of LDAP Search Filters." Table 1 lists a few examples of some common search filters.

Filter String	Description
(&(objectClass=person(!(uid=sharber)))	All person objects except Steve Harber
(&(objectClass=person(|(sn=Thomas)(sn=Miller)))	All pserons with a surname of Thomas or Miller
(&(ou=People)(objectClass=person))	All persons in the People Organizational unit
(sn=da*)	All objects with a surname that starts with da
Objectclass=person	All records where the Objectclass is a Person

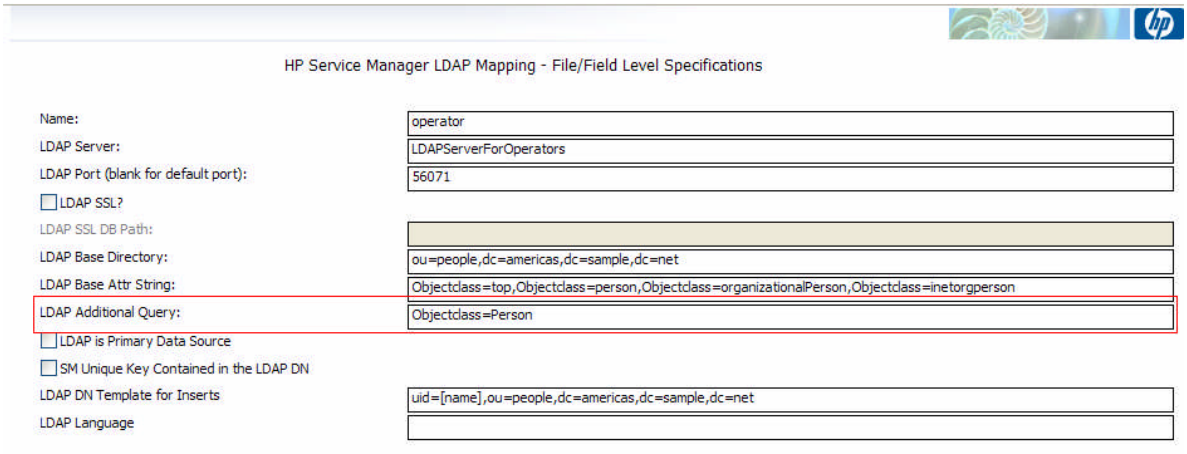
Table 1: Common LDAP search filters

Example: Limiting the search by setting an additional query:

The following section shows that BOB.HELPDESK is in the Object class=person.



To search only for records in that object class, set up the scldapfile record as follows:



The ldaptimelimit parameter

The ldaptimelimit parameter is used to specify the number of seconds that Service Manager waits for a response to an LDAP query before canceling the request. This parameter is often used to prevent performance issues that could result from an inefficient query to the directory server. The format of the ldaptimelimit parameter is

```
ldaptimelimit:<seconds>
```

Where *seconds* specifies the maximum amount of time Service Manager should wait. If this parameter is not used, or if *seconds* is set to 0, no time limit is set.

Note: If a time limitation is encountered, an "**LDAP Error 3 Time Limit Exceeded**" message is received. This error is mainly informational. However, the completeness of the data returned from the directory server cannot be certified if the error is encountered. Use of the ldapsearchscope parameter and filtering methods may prevent these errors.

The ldapmaxrecords parameter

The ldapmaxrecords parameter specifies the maximum number of records that the directory server will return as a result of a query. As with the ldaptimelimit parameter, the ldapmaxrecords parameter can prevent the processing of inefficient queries. For example, if a user mistakenly issues a true query on a table that contains hundreds of thousands of records, the query stops after encountering the maximum record limit to prevent performance problems.

Note: If a record limitation is encountered, an "LDAP Error 4 Size Limit Exceeded" message is received. This error is mainly informational. However, the completeness of the data returned from the directory server cannot be certified if the error is encountered. Use of the ldapsearchscope parameter and filtering methods may prevent these errors from occurring.

Your directory server administrator can set both a time limit and a maximum records parameter for the directory server. This can be misleading because time and size limits can be set both on the LDAP server and the LDAP client (i.e. Service Manager. For example, your Service Manager system may not be using either parameter, but your Active Directory system is set to its out-of-box limit of 1000 records. Therefore, you are limited to 1000 records, and any attempt to retrieve more records causes the "LDAP Error 4 Size Limit Exceeded" message. If you receive either of these messages, contact your directory server administrator to determine the proper settings for both systems.

In terms of the Active Directory system described above, the restricted parameter is called *MaxPageSize*. This value can be changed using the ntdsutil.exe program supplied with Windows 2000 Server. Another method to change this parameter is to edit it directly inside the CN=Default Query Policy, CN=Query-Policies, CN=Directory Service, CN=Windows NT, CN=Services, CN=Configuration, DC=YOUR_COMPANY, DC=YOUR_COMPANY_TLD entry by using an LDAP

administrator tool. In both cases, you must have administrator rights. For more information, consult your LDAP administrator.

Implementing your directory server integration

After you have completed the planning and final design of your directory server integration, the next step is to move into the actual implementation. HP Software recommends that integration first be implemented on a development or test system and moved to your production system *only* after it has been thoroughly tested and approved. This can prevent performance failures or security issues that may result in system down time.

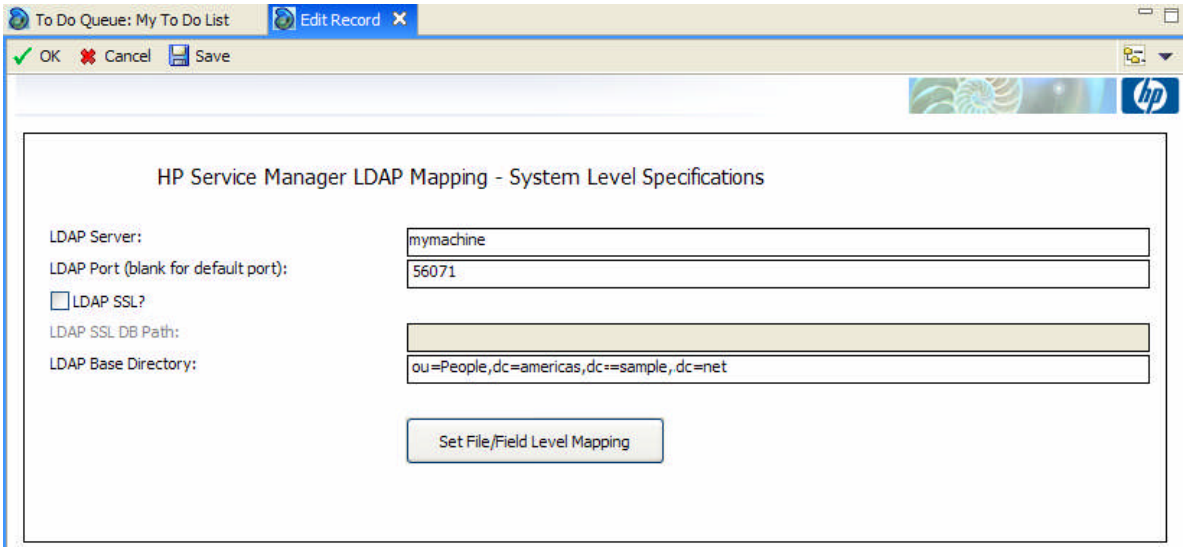
Implementation of your integration involves the following steps:

- Configuring the Service Manager LDAP system level interface
- Defining the Service Manager file LDAP mappings
- Enabling the appropriate Service Manager parameters

Note: The following steps describe a basic Service Manager Directory Server integration. If you are integrating with a directory server that uses an SSL connection, or if you plan to allow insertions into your directory server, you should first configure the interface with a non-SSL port to establish connectivity and ease troubleshooting.

Configuring the Service Manager LDAP system level interface

The Service Manager LDAP system level interface is defined in the scldapconfig record. To modify it click **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **System** – **LDAP Mapping**. Unless you are using a SSL LDAP connection or a port other than the default port of 389 for regular LDAP connections or 636 for LDAP with SSL you only need to fill in the LDAP Server and LDAP Base Directory fields.



The LDAP server field should contain the name of your LDAP server. This can be either the machine name or its address, as long as your Service Manager server can resolve the value. Test the connection by pinging the value you entered in the LDAP Server Field. If you do not see successful responses, the value will not work. Contact the directory server administrator to acquire the correct value.

If your directory server does not connect via the standard LDAP ports (389 for non-secure connections and 636 for secure connections), enter the correct port number in the LDAP port field.

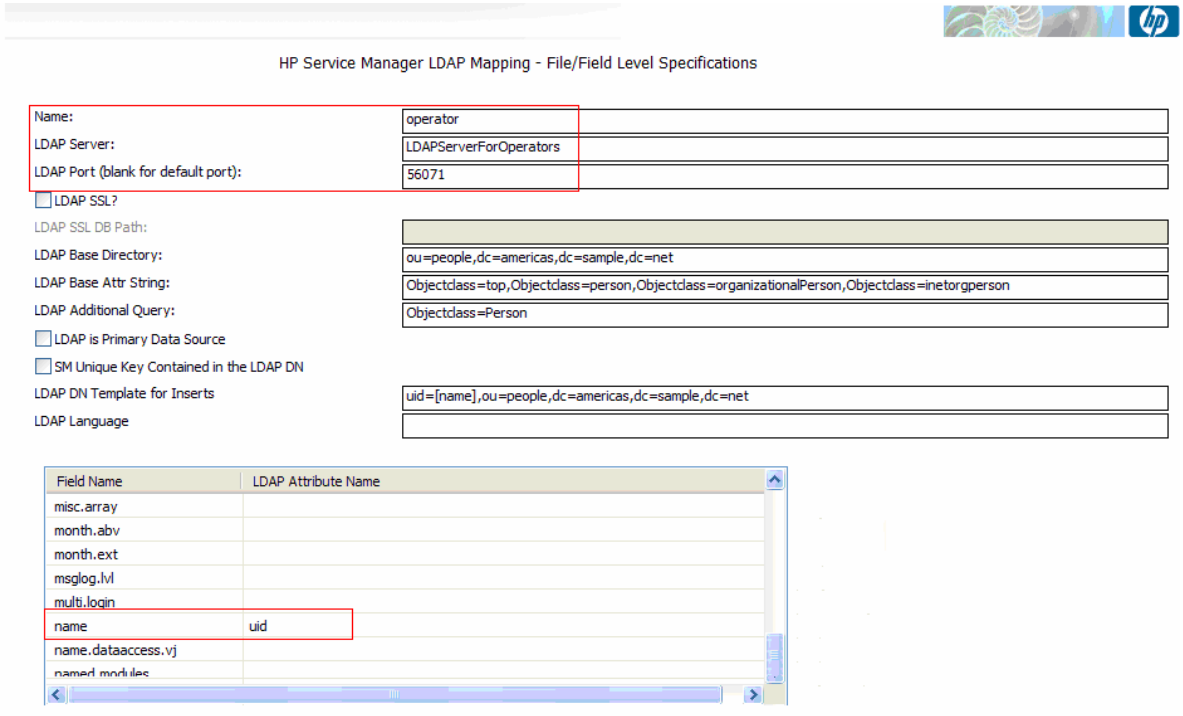
The LDAP Base Directory field should be set as previously discussed in the section "Setting the correct base DN"

After you enter these basic fields, click **Set File/Field Level Mappings** and define your LDAP file level mappings in the scldapfile table for each table that is supposed to be mapped to LDAP.

Defining the Service Manager file LDAP mappings

Service Manager can be mapped to multiple directory servers on a per file basis as depicted in Figure 11 below. In this example, the company's customer data is located on a Sun ONE directory server controlled by the Sales Department; their device data is located on a Microsoft Active Directory server in the IT Department; and their employee data is located on a Lotus Domino server maintained by the Human Resources Department.

Through Service Manager file level mappings, the company can integrate to each of these directories by simply completing the LDAP server level information located in the scldapfile table. The scldapfile table contains a record for each Service Manager file. Each record contains an array of the fields in the file so that they can be individually mapped to directory server attributes.



Note: If the directory server interface information is set in the scldapfile record of a table, Service Manager uses this to connect to the directory server and ignores the information that is set in the scldapconfig record. The scldapconfig record can still be used as a default configuration for other files.

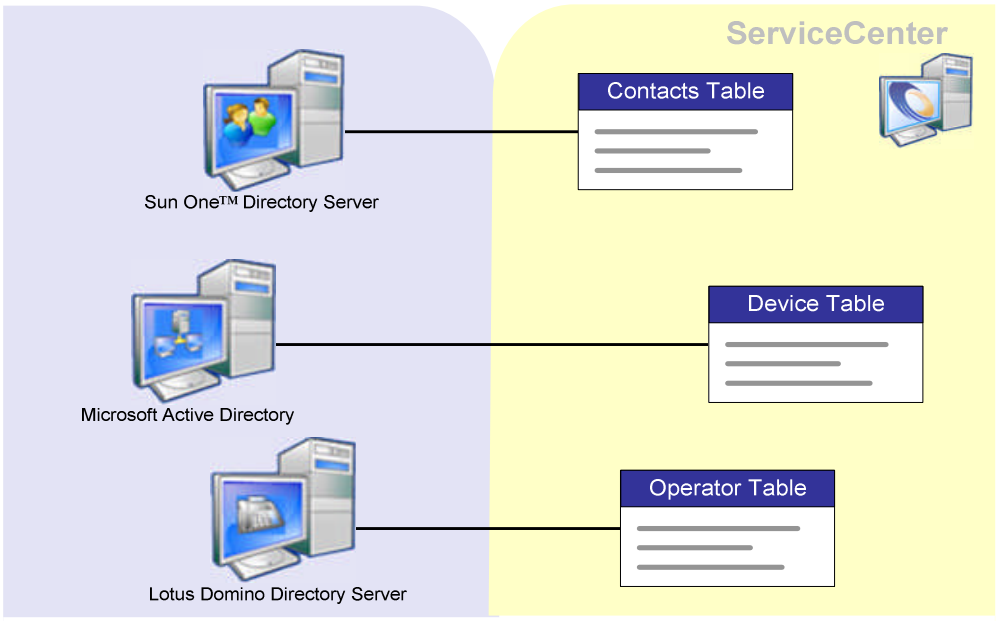


Figure 11 Integrating Service Manager with multiple directory servers

First, open the record for the Service Manager file you want to map to a directory server. If you are using multiple directory servers, enter LDAP system-level information in the appropriate fields for each related file, such as LDAP Server, LDAP Port, and LDAP Base Directory. If these fields are left blank, they default to the values set in the `scldapconfig.g` record. Be sure to verify your connection information as detailed in the previous section, *Configuring the Service Manager LDAP system level interface*.

To configure the field for attribute-level mapping, locate the field name array and enter the corresponding directory server field into its LDAP Attribute Name field. To improve performance and limit network traffic, map *only* the fields you need. Ensure that these values are protected appropriately on the directory server. For example, if phone numbers are mapped from your directory server to the Service Manager contacts file, ask yourself whether you want the value to be updated by anyone who has access to it in Service Manager. If not, ensure that the user who is binding to the LDAP server does not have write access to the item in the directory server. Conversely, if there are specific users who make modifications that need to be propagated to the directory server, their accounts on the directory server need to have the appropriate access rights.

After defining your field mappings, you set the LDAP is Primary Data Source as well as the LDAP Additional Query field if required in your design. See the section *Planning your LDAP integration*.

Note: The Service Manager operator file contains a password field. There is no need to map this field to an attribute in the directory server. For security reasons, no directory server allows password information to be retrieved from an LDAP query. Service Manager ignores any attributes you have mapped to the password field, and does not store the directory server password value in the `operator` file. The only exception is when the password is supposed to be changed on LDAP from Service Manager, the password field needs to be mapped.

Service Manager LDAP parameters

Table 2 below lists all of the available Service Manager parameters that you can use for tailoring your LDAP interface. The parameters should be placed in your `sm.ini` file. Some of them can be used as command-line options when starting the Service Manager server.

Note: It is not necessary to restart the Service Manager server after making changes to these parameters in your `sm.ini` file. However, all parameter chages in the sm.ini as well as changes to scldapconfig and scldapfile require that the client be restarted.

Parameter	Usage	Valid Releases
ldapauthenticateonly	ldapauthenticateonly	Added in 5.0.3.7
	Description	
	LDAP is used for authentication only • Service Manager disconnects from LDAP after the user is authenticated. • LDAP File Mappings are ignored. • Prior to Service Manager versions 5.1.5, 6.0.1, and 6.1.0 this parameter could not be used when LDAP was set as the primary data source.	
Parameter	**Usage**	**Valid Releases**
ldapbinddn **ldapbindpass**	ldapbinddn:<*DN of the LDAP bind account*> ldapbindpass:<*password for the LDAP bind account*>	Added in 4.0
	Description	
	Service Manager uses this information to initially bind to the directory server. • Most often used when the directory server does not allow anonymous binds. • Should be used for data warehousing without authentication. • The DN needs to match the directory server account. • If used during authentication, the bind account will be replaced with the logged in user after the user is authenticated.	
Parameter	**Usage**	**Valid Releases**
ldapdisable	ldapdisable:0 ldapdisable:1	All Releases
	Description	
	0 The LDAP interface is connected. (Default) 1 The LDAP interface is disconnected and ignored. • Useful during implementation of the directory server interface • A valuable debugging tool	
Parameter	**Usage**	**Valid Releases**
ldapdisconnect	ldapdisconnect:0 ldapdisconnect:1	Added in 5.1.2.0
	Description	
	0 Service Manager maintains its connection to the directory server throughout the client session. (Default) 1 Service Manager connects to and disconnects from the directory server for each LDAP transaction. • Often used to eliminate an open network port that is considered a security risk. • Use of this parameter increases network traffic. • LDAP administrators are often wary of leaving an open port connected to the directory server. Enabling this parameter prevents that.	

	Note: The ldapdisconnect parameter is not necessary when the ldapauthenticateonly parameter is used because the connection is closed immediately after authentication.	
Parameter	**Usage**	**Valid Releases**
ldapmaxrecords	ldapmaxrecords:<*record limit*>	Added in 5.0.0.0
	Description	
	• Should be set to a reasonable amount that ensures the correct query result set. • Often used to prevent performance problems that are caused by inefficient or incorrect queries. • If directory server has a maximum record limit less than <*record limit*>, this parameter is not necessary and is ignored. • Service Manager defers to the directory server limitation. • The default value is no limit.	
Parameter	**Usage**	**Valid Releases**
ldaptimelimit	ldaptimelimit:<*seconds*>	Added in 5.0.0.0
	Description	
	0 No time limit (Default) • Should be set to a reasonable number of seconds that ensures the correct query result set. • Often used to prevent performance problems caused by inefficient or incorrect queries. • If directory server has a time limit less than <*seconds*>, this parameter should not be used • Service Manager defers to the directory server limitation.	
Parameter	**Usage**	**Valid Releases**
ldapnostrictlogin	ldapnostrictlogin:0 ldapnostrictlogin:1	Added in 5.1.2.0
	Description	
	0 The user must have a valid directory server account and password to log into Service Manager. (Default) 1 The user is not required to have a valid directory server account to log into Service Manager. • Often used to allow connections to Service Manager by contacts who are not contained in the company's directory server. • Prior to Service Manager 6.0, this parameter could not be used when LDAP was set as the primary data source.	
Parameter	**Usage**	**Valid Releases**
ldapstats	ldapstats:0 ldapstats:1	All Releases
	Description	
	0 Disable printing of LDAP Messages to the `sm.log` file. (Default) 1 Enable printing of LDAP Messages to the `sm.log` file.	

	• Should be enabled only when debugging an LDAP- related issue.	
Parameter	**Usage**	**Valid Releases**
ldapsearchscope	ldapsearchscope:0 ldapsearchscope:1	Added in 6.0.0.0
	Description	
	0 The scope of LDAP queries is set to base directories and all sub trees. (Default) 1 The scope of LDAP queries is set to the base directory only.	
Parameter	**Usage**	**Valid Releases**
ldapserver	ldapserver<*n*>:<*hostname*>,<*port*>,<"*base directory*">,<*certificate file path*>,<*key file path*>	Added in 5.1.2.0
	Description	
	<*n*> A number greater than or equal to 1 that represents the failover order of the server. <*hostname*> The host name of the LDAP server (Required) <*port*> ***optional*** The communications port on which the LDAP server listens for connection requests. The default is 389. <"*base directory*"> ***optional*** The base directory on the directory server where LDAP searches begin. Quotation marks are required around the base directory path. <*certificate file path*> ***optional*** The complete path to the directory server SSL certificate <*key file path*> ***optional*** The complete path to the directory server client authentication SSL certificate • Used to configure failover processing for the directory server configuration. • Parameter is not required if using an LDAP proxy server.	

Configuring the Service Manager LDAP interface for SSL

Because security is a top priority, directory servers can be configured to communicate over SSL connections that protect networks from outside attacks and secure proprietary information. Secure Socket Layer (SSL) is a protocol that lets servers and clients communicate more securely through encryption. Without SSL, data sent between the client and server is vulnerable to packet sniffing by anyone with physical access to the network. SSL provides data encryption, server authentication, message integrity, and client authentication over the server-to-client connection. Figure 12 describes the SSL handshake process that takes place prior to connection.

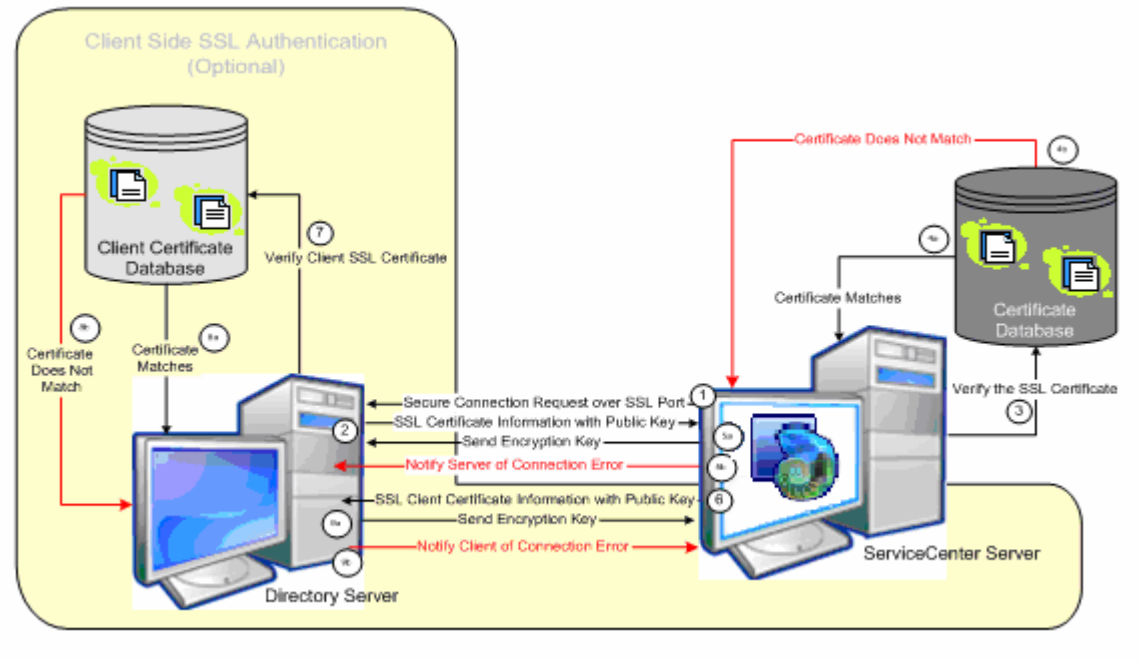


Figure 12: SSL handshake process

SSL can be configured to validate Service Manager connections on both the server side and the client side. Most companies set up SSL authentication only on the server side, but others add client-side authentication for a higher level of security.

Note: Server Side authentication puts a certificate only onto the LDAP server, client side authentication adds a certificate on the Service Manager server that needs to be known to the LDAP server as well.

Configuring the server-side SSL connection

For server-side SSL authentication, you store a copy of the directory server's SSL certificate on the Service Manager server, and then use that certificate to verify the security information that the directory server sends when a connection request is issued. After following the steps in this section, proceed to Configuring the Service Manager LDAP Interface for setting up the Service Manager LDAP interface.

Example: Setting up Server Side SSL authentication on Sun ONE Directory Server on Windows

To set up server side SSL authentication, a signed certificate has to be added to the LDAP configuration and the connection information in Service Manager has to be updated to connect to the LDAP secure port instead of the default port. Follow these steps to set up LDAP SSL for Sun ONE and Service Manager.

1. Set up OpenSSL

   The openssl tools are required to process certificate requests. You can download the source code at http://www.openssl.org and build the tools. OpenSSL version 0.9.8d is included with SM7 and can be used as well.

   The following steps allow you to use the openssl executable in the Service Manager RUN directory.

a. Set an environment variable called OPENSSL_CONF with the fully qualified file name of the openssl.conf file (e.g. C:\Documents and Settings\<user id>\openssl\openssl.conf). Alternatively, add –config <path to the openssl.conf file> to the end of the openssl commands

2. Create a private key file cakey.pem in the <*path*> directory

   a. Enter the following command in your openssl directory

```
openssl genrsa -des3 -out <path>\cakey.pem 4096
```

   b. When prompted for password, enter a password and confirm it. Note this password, as the cakey pass phrase will be needed regularly.

3. Create a self signed CA certificate

   a. Enter the following command in your openssl directory

```
openssl req -new -x509 -days 999 -out <path>\cacert.pem
```

   b. When prompted for the cakey pass phrase password enter the password noted in step 2b

   c. When prompted, enter the information from the sample below:

```
     - Country                 [e.g. US]
     - State                   [e.g. California]
     - City/Locality           [e.g. San Diego]
     - Organization            [e.g. HP]
     - Organizational Unit     [e.g. SM R&D]
     - Common name             [e.g. Jane Doe]
     - Email address           [e.g. your SEA]
```

   The self signed CA certificate is now in <*path*>\cacert.pem.

4. Create a certificate request

   a. Open Sun ONE Console

   b. Login as admin

   c. Go to the Directory Server

   d. Open the Directory Server and click on the **Tasks** tab

   e. Click on Manage Certificates and if prompted, enter a password for the internal (software) token.

   f. On the **Server Certs** tab click the **Request...** button

   g. In the next window, set the option to **'Request certficate manually'**. Click **Next.**

   h. Enter this information when prompted:

```
Server name (e.g. mymachine.americas.sample.net - must be pingable)
Organization (usually company name, e.g. HP),
Organization unit (department name, project name, area, for example
ETM,
City/Location (e.g. San Diego),
State (e.g. California, usually not the 2 letter acronym),
Country (e.g. US, this again should be the official country domain)
```

   Enter at a minimum Organization, State and Country. Click **Next**.

   i. Enter the password for the internal (software) token from step 4e. Click **Next**

   j. Save the certificate request to a file and call it for example ldap.<host name>.certificate.request.pem

   k. Click **Done.**

5. Create the certificate from the certificate request

   a. Copy the file containing Sun ONEs certificate request into the openssl directory

   b. Open the certificate request in an ASCII text editor, remove all empty lines from the ldap.<*host name*>.certificate.request.pem file

   c. Enter the following command

```
openssl ca -days 999 -in ldap.<host name>.certificate.request.pem -out
ldap.<host name>.certificate.pem
```

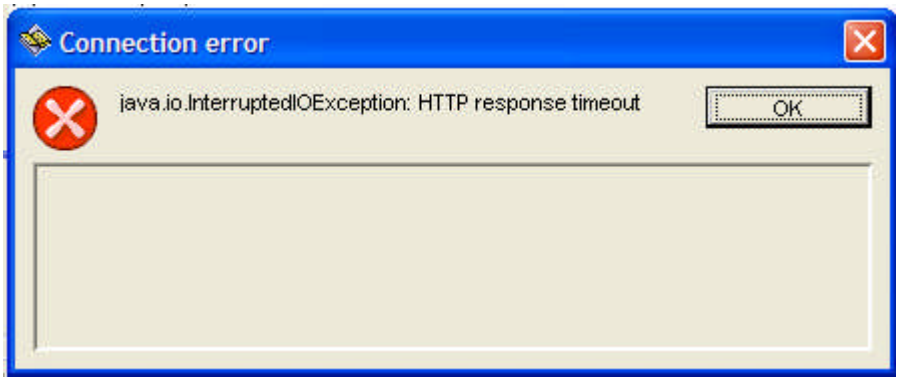
   d. When prompted for the cakey.pem password enter the password set in step 2-b

   e. When prompted whether you want to sign the certificate, enter **y**

   f. When prompted whether you want to commit the signature, enter **y**

   You now have a signed certificate in file ldap.<*host name*>.certificate.pem

Troubleshooting Note: With Sun ONE it seems to be necessary to restart both LDAP services (admin and directory server) and the Admin Console UI every time you get the error message pictured below. Check after stopping the services that the security.exe process is not found in Task Manager, otherwise kill the process before restarting the services.



6. Install the CA certificate

   a. Copy the certificates ldap.<*host name*>.certificate.pem and cacert.pem back to your Sun ONE machine.

   b. Edit the ldap.<*host name*>.certificate.pem to contain only the following:

```
-----BEGIN CERTIFICATE-----
[certificate information]
 -----END CERTIFICATE-----
```

   Remove empty lines if exists.

   c. In Sun ONE click on **Manage Certificates** (see steps 4)

   d. Click on the **CA Certs** tab

   e. Click on **Install**

   f. Click on **In this local file** and point to the cacert.pem . Click **Next**

   g. When displayed information about your self signed CA certificate, click **Next**.

   h. (Step applies only to Sun ONE versions prior to 6.x) When asked for the name of certificate. click Next.

   i. (Step applies only to Sun ONE versions prior to 6.x) When asked for the purpose of the certificate, verify that both client and server authentication are enabled (Default in version 6.x).

j. Click **Done**.

k. After installing the cacert.pem in the list of CA Certs, verify that your certificate is in the list on the CA Certs tab.

7. Install the Server certicate

   a. In the **Server Certs** tab, and click on **Install**.

   b. Select **In this local file** and point to ldap.<*host name*>.certificate.pem . Click **Next**

   c. When displayed information about the certificate, click **Next**

   d. When prompted for the name of the certificate, use the default: **server-cert**, then click **Next**

   e. Enter the password for the internal (software) token and click **Done**

   f. Verify that your certificate displays in the Server certs tab

   g. Click **Close**

8. Configuring the LDAP server to enable SSL

   a. In the Directory Server UI, select your LDAP server, then go to the **Configuration** tab

   b. Click on the **Encryption** sub tab

   c. Check the **Enable SSL for this server** flag

   d. Check the **Use this cipher family: RSA** flag

   e. Select **Do not allow client authentication** or **Allow client authentication**. Click on **Save**.

   f. When prompted that you need to restart the directory server, click **OK**

   g. On the **Tasks** tab click **Restart Directory Server**, when prompted confirm with **Yes**.

   h. On startup of the directory server, enter the password for the internal (software) token

Additional Steps for Active Directory Users

Active Directory requires at least two certificates. The openssl utility creates all but one certificate for the Active Directory server. To generate the other certificate, perform the following additional steps:

1. Locally log in to the Active Directory machine.
2. Click **Start** -> **Programs** -> **Administrative Tools** -> **Certification Authority**.
3. Locate your local root certificate authority (CA). Search under the "Certification Authority (Local)" entry in the tree on the left side panel.
4. Right-click the local root CA entry and click **Properties**.
5. Click the **General** tab and then click **View Certificate**.
6. Click the **Details** tab and click **Copy to File** to open the Certificate Export wizard.
7. Click **Next** to open the Export File Format page.
8. Click to select the **DER encoded binary X.509 (.CER)** format and click **Next**.
9. Enter a new filename where your exported certificate will be stored and click **Next**.
10. Review your settings and click **Finish**. A message box indicates that the export operation was successful.
11. Terminate the Certificate Authority utility and copy the exported certificate to the `RUN` directory of your Service Manager server.
12. Open a command window from the RUN directory and enter the following command:

```
openssl x509 –inform DER –in <your export file>
```

13. The final certificate is created and placed between a **("-----BEGIN CERTIFICATE-----")** header and a **("-----END CERTIFICATE-----")** footer.
14. Copy this certificate, including the header and footer, to the end of your `trusted.certs.pem` file, below the first certificate you generated.

Configuring the client-side SSL authentication

For client-side SSL authentication, a copy of the Service Manager client certificate must reside on the directory server, which validates Service Manager security information after server-side SSL authentication has been verified. The certificate you create must be signed by a certification authority (CA) before it can be used for SSL communications. System administrators can either set up their own CA or use an outside company such as VeriSign® to sign their certificates.

This document will only discuss the steps that are done on the Service Manager system. All steps on the LDAP system must be performed by the LDAP administrator and are documented procedures on these LDAP servers. Refer to the LDAP server documentation for more information.

Note: Client-side SSL authentication is an optional configuration. Contact your directory server and your security administrator to determine whether client-side SSL authentication is necessary.

Creating the Service Manager client certificate

Follow these steps to generate your client key file:

1. Create a new text file called `openssl.cnf`, which will be the certification authority configuration file. Save the file to your Service Manager `RUN` directory.
2. Add the text listed below to the `openssl.cnf` file. (This file can also be found on the OpenSSL web site at http://www.openssl.org.)

```
HOME        = .
RANDFILE    = $ENV::$HOME/.rnd
oid_section    = new_oids

[ new_oids ]

[ ca ]
default_ca     = CA_default         # The default ca section

[ CA_default ]
dir         = ./demoCA        # Where everything is kept
certs       = $dir/certs            # Where the issued certs are kept
crl_dir        = $dir/crl                # Where the issued crl are kept
database    = $dir/index.txt        # database index file.
new_certs_dir     = $dir/newcerts         # default place for new certs.
certificate    = $dir/cacert.pem    # The CA certificate
serial         = $dir/serial        # The current serial number
crl         = $dir/crl.pem          # The current CRL
private_key    = $dir/private/cakey.pem   # The private key
RANDFILE    = $dir/private/.rand    # private random number file
x509_extensions      = usr_cert           # The extensions to add to the
cert
name_opt       = ca_default         # Subject Name options
cert_opt          = ca_default            # Certificate field options
default_days      = 365            # how long to certify for
default_crl_days     = 30                # how long before next CRL
default_md     = md5          # which md to use.
Preserve    = no        # keep passed DN ordering
policy         = policy_match

[ policy_match ]
```

```
countryName    = match
stateOrProvinceName  = match
organizationName  = match
organizationalUnitName  = optional
commonName     = supplied
emailAddress      = optional

[ policy_anything ]
countryName       = optional
stateOrProvinceName     = optional
localityName         = optional
organizationName     = optional
organizationalUnitName        = optional
commonName        = supplied
emailAddress         = optional

[ req ]
default_bits         = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes        = req_attributes
x509_extensions         = v3_ca
string_mask       = nombstr
[ req_distinguished_name ]
countryName       = Country Name (2 letter code)
countryName_default     = US
countryName_min      = 2
countryName_max      = 2
stateOrProvinceName     = State or Province Name (full name)
stateOrProvinceName_default   = Some-State
localityName         = Locality Name (eg, city)
0.organizationName      = Organization Name (eg, company)
0.organizationName_default    = Internet Widgits Pty Ltd
organizationalUnitName        = Organizational Unit Name (eg, section)
commonName        = Common Name (eg, YOUR name)
commonName_max    = 64
emailAddress         = Email Address
emailAddress_max     = 64

[ req_attributes ]
challengePassword    = A challenge password
challengePassword_min         = 4
challengePassword_max         = 20
unstructuredName     = An optional company name

[ usr_cert ]
basicConstraints        =CA:FALSE
nsComment         = "OpenSSL Generated Certificate"
subjectKeyIdentifier    =hash
authorityKeyIdentifier        =keyid,issuer:always

[ v3_req ]
basicConstraints         = CA:FALSE
keyUsage          = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]
subjectKeyIdentifier    =hash
authorityKeyIdentifier        =keyid:always,issuer:always
basicConstraints         = CA:true

[ crl_ext ]
```

```
authorityKeyIdentifier        =keyid:always,issuer:always
```

3. Create an SSL client key file and a client certificate request by executing the following from a command window:

```
openssl req –newkey rsa:1024 –nodes –keyout client.key.pem –out
client.request.pem –config openssl.cnf
```

4. You are prompted to enter a PEM password, which is a value that you choose as a password to your certificate. You are also prompted to enter information about your client (your Service Manager server) connection. The OpenSSL utility output is shown below:

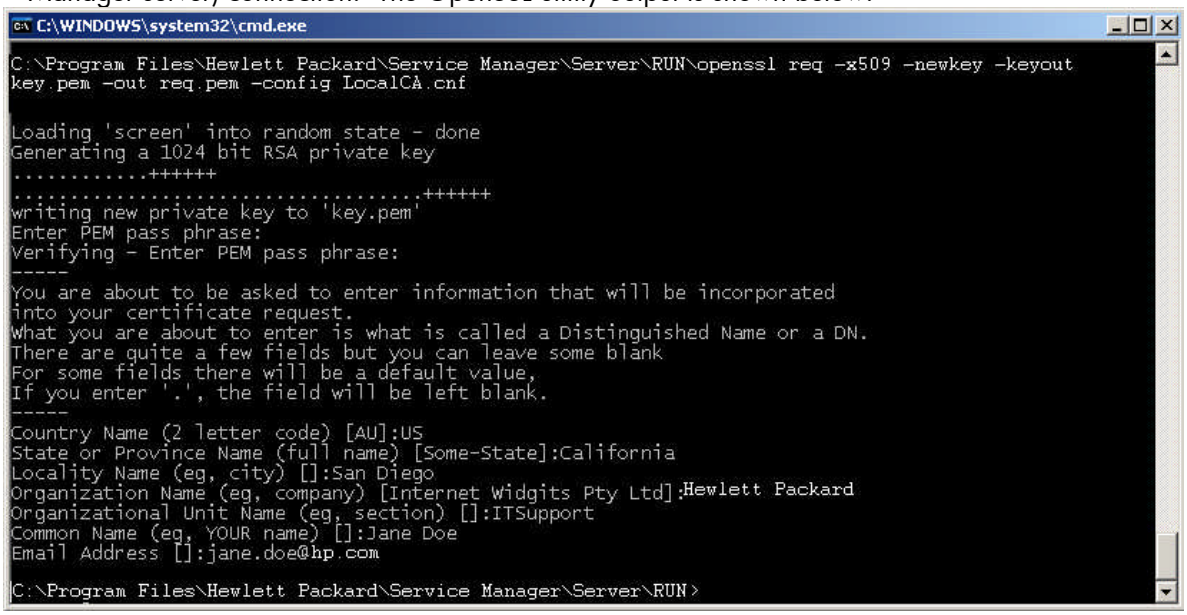


Figure 13: Output from the openSSL req command

5. After you have created your client certificate request, you get it signed by a certification authority or use your own cacert for signing as in the example for server side ssl authentication above and have it stored in file called `client.certificate.pem` on the LDAP server.
   Note: This process varies from LDAP server to LDAP server and your LDAP system administrator should be available to install the client certificate on your LDAP server. Refer to your LDAP server's documentation for more information.
6. Create or update the `ldaprc` file, which overrides global LDAP values and sets the certificate and private key used to establish client authentication. If the file does not already exist, create a text file named `ldaprc` in the Service Manager `RUN` directory and add the following lines of text to it.

   TLS_CERT *<filename of client certificate file>*

   filename of client certificate file is the `client.certificate.pem` file in step 5 in the example above.

   TLS_KEY *<filename of the client key>*

   filename of the client key is the `client.key.pem` file in step 3 in the example above.

   Note: It may be necessary to specify fully qualified filenames. For more information about the `ldaprc` file and the use of the TLS commands, go to http://www.openldap.org/pub/ksoper/OpenLDAP_TLS_howto.html#5.2.
7. When you restart your Service Manager client, the client SSL authentication changes take effect.

Configuring the Service Manager LDAP Interface

First you will need to gather your trusted certificates, then setup your scldapconfig record to use the SSL connection.

Gathering your trusted certificates

To use SSL with the Service Manager LDAP interface, you need an ASCII text file that contains all your trusted SSL certificates.

Follow these steps to generate your trusted certificates file, `trusted.certs.pem`:

1. Execute the following command from your Service Manager server's `RUN` directory to generate a list of all the certificates required to connect to your directory server
   `openssl s_client –connect <LDAP hostname>:<LDAP SSL port> -showcerts`
2. This utility generates a lot of output and requires that you type **CTRL**+**C** once to exit. There is at least one certificate in the output (depending on your LDAP server more), each of which is located between a **("-----BEGIN CERTIFICATE-----")** header and a **("-----END CERTIFICATE-----")** footer. Copy these certificates, including their headers and footers, into the `trusted.certs.pem` file.

Note: This is not the only way to generate the certificates required to configure your system with SSL. Check with your directory server administrator to determine your company's preferred method.

Configuring the Service Manager LDAP Interface

Follow these steps to configure the Service Manager LDAP interface:

1. Log in to Service Manager as a system administrator.
2. Click **Menu Navigation** – **System Administration** – **Ongoing Maintenance** – **System** – **LDAP Mapping** to open the LDAP System Level Configuration form, scldapconfig.g.
3. If you are connecting to a single directory server, populate the following values in the scldapconfig record. If you are connecting to multiple directory servers, leave the scldapconfig record empty and click **Set File/Field Level Mapping** to skip to the Service Manager LDAP Mapping Form, scldapfile.g.
4. If you have already set up the LDAP interface to work with a non-SSL connection, your LDAP Server and LDAP Base Directory information should already be in place. Populate the following fields using the appropriate form as discussed in the previous step.

   Note: Ensure to use the exact same machine name here as was used in the certificate – typically the fully qualified name.

LDAP Port	Modify to indicate the SSL port (default is 636)
LDAP SSL?	Check to enable
LDAP SSL DB Path	Fill with fully qualified name of the trusted certificate file, e.g. `c:\program files\hp\service manager\server\run\trusted.certs.pem`

5. Restart your Service Manager client for your changes to take effect.

Configuring Service Manager to insert objects into the directory server

It may be necessary to add newly created Service Manager data into your directory server. If configured correctly, Service Manager can insert data on the Directory Server by sending an LDAP add request to the directory server whenever a record is added to a file that is mapped to the

directory server. To add an object, the directory server needs to know exactly where in the directory to place it and may require that certain attributes be included when inserting a new object to ensure that all objects contain a consistent dataset.

Verifying the directory server access rights

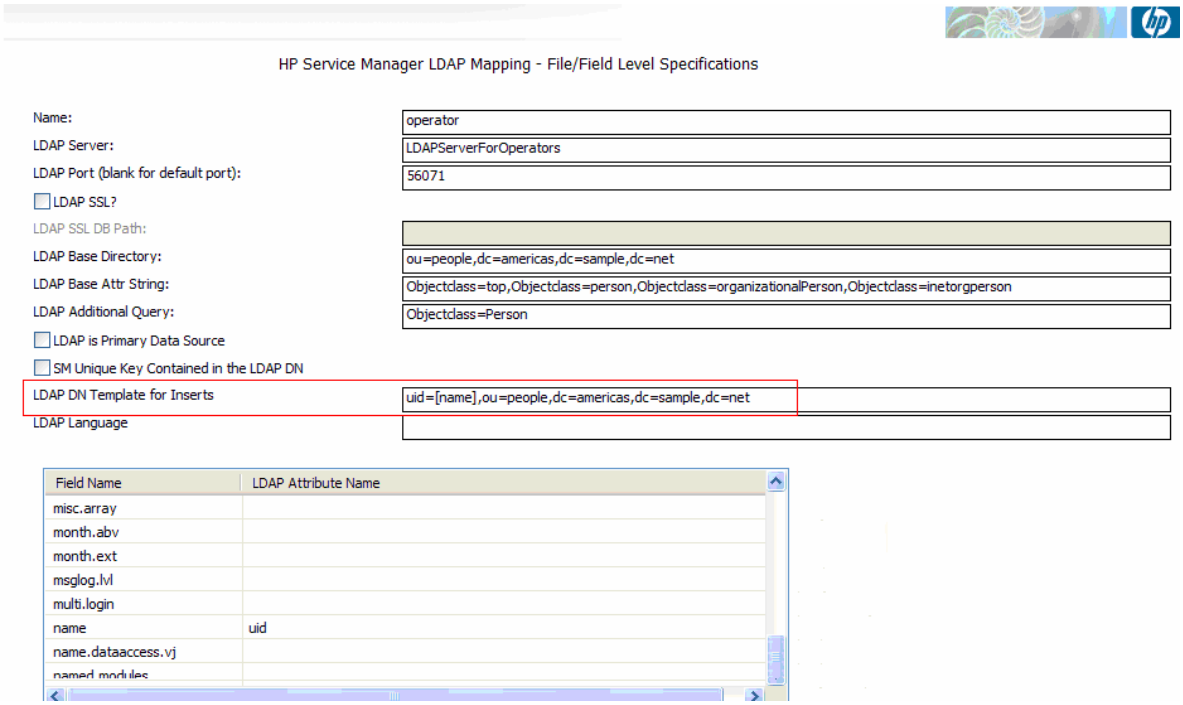
To add new objects to a directory server, the appropriate access rights must be in place for the bind account user. If these rights are not in place, the add request fails and the data between Service Manager and the directory server become inconsistent. Discuss the directory server's access rights and account settings with your LDAP administrator to ensure they are set correctly.

LDAP DN Template for Inserts

The LDAP DN Template for Inserts field in the scldapfile record allows objects to be inserted based on field data from the Service Manager file. The field has to be a complete DN for example **uid=falcon,ou=people,dc=Americas,dc=sample,dc=net.**

The field can be constructed as:

LDAP attribute1=[*SC fieldname1*],LDAP attribute2=[*SC fieldname2*],LDAP attribute 3=[*SC fieldname3*],...



Note: The LDAP DN Template for Inserts field can contain attribute and fieldname or attribute and value pairs. They can contain as many attributes as necessary. Figure 13 depicts an example insertion that originates from the Service Manager `contacts` file:

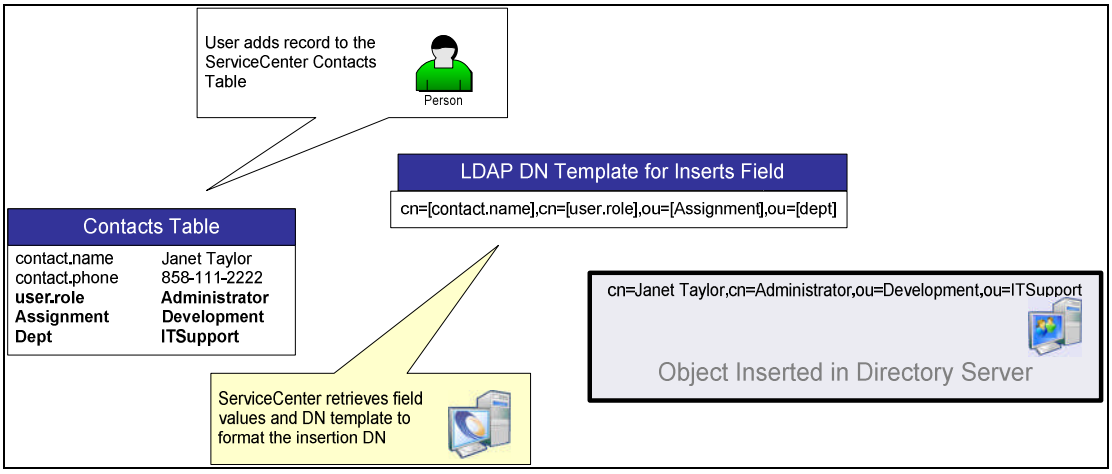


Figure 13: Using the LDAP DN Template for Inserts field

The LDAP DN Template for Inserts field is very flexible and can be configured so that it uses a field defined in the Service Manager table that contains the actual directory location as depicted in Figure 14:

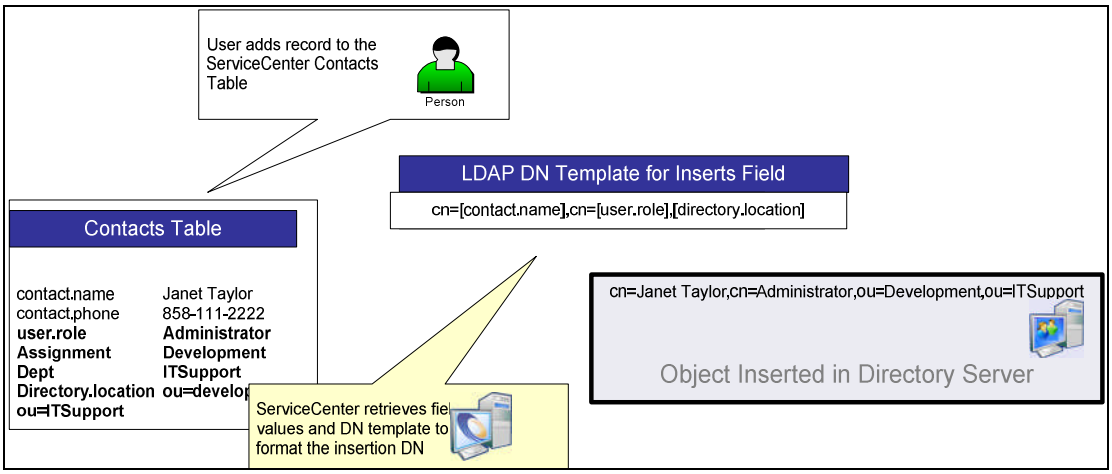
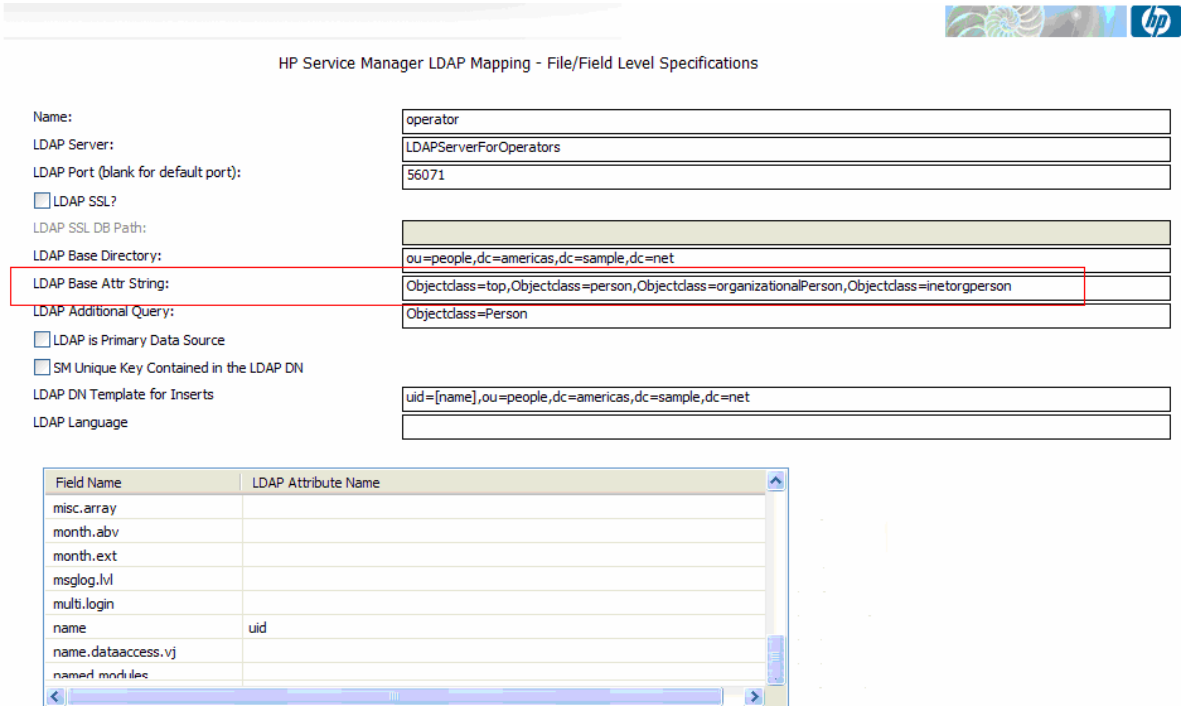


Figure 14: Using LDAP DN Template with Field Location

Handling required attributes

If your directory server is configured to require certain attributes when creating an object, use the LDAP Base Attr String field located in the LDAP File Mapping form, scldapfile.g. This field contains a string of required attributes and their values. This string is appended to the LDAP Add request that Service Manager sends to the directory server. If a required attribute is already mapped to Service Manager via the Fieldname array, it does not need to be included in this field. For example, if your Sun ONE directory server requires that the objectClass attribute be included when inserting a new object, add the following value to the LDAP Base Attr String field:

objectClass=top,objectClass=person,objectClass=organizationalPerson, objectClass=inetorgperson

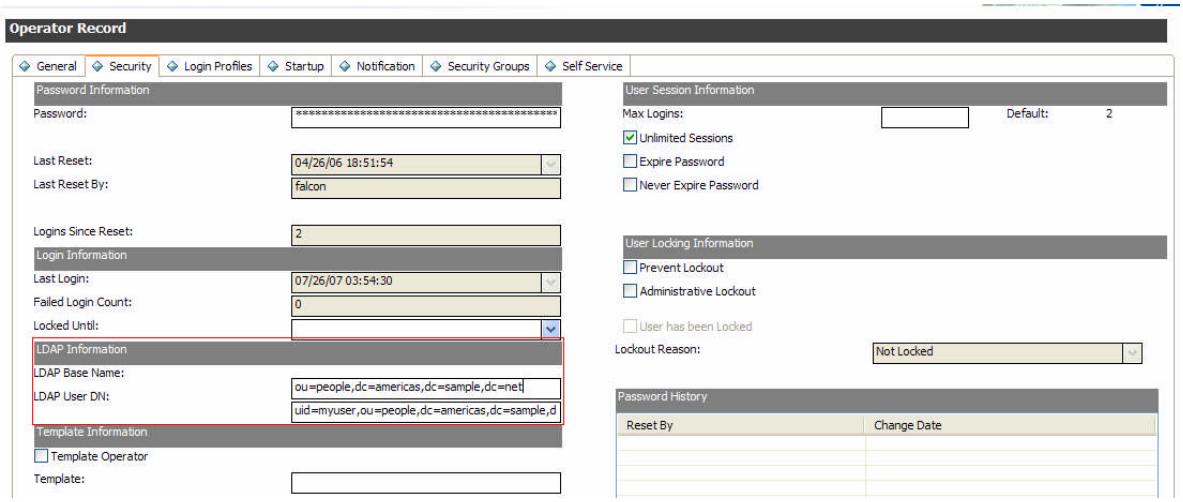


Using the Service Manager operator record to configure the LDAP interface

The Service Manager operator file contains fields that can be used to configure the LDAP interface. Modifications to the operator record affect only that user, and may be preferable to making changes that affect the entire LDAP interface.

Limiting access via the LDAP Base Name field

The LDAP Base Name field, which is found on the Security tab of the operator record in the LDAP Settings section, is available in all versions of Service Manager. It prevents an operator who logs into Service Manager from being authenticated if the operator is not located in the LDAP Base Name location of the directory server.



Note: This field can cause inconsistent results if used when LDAP is set to be the primary data source. HP Software recommends that you exercise caution when setting this field.

Binding without mapping the operator table

Note: This applies only to ServiceCenter 6.0.2 and later versions.

The LDAP User DN field was added to the operator file in ServiceCenter 6.0.2. The LDAP User DN field contains a string that represents the DN Service Manager uses to bind the user to the directory server. It allows Service Manager to authenticate using a directory server without mapping the primary key or any field of the operator file to LDAP. With this setting, the connection to the LDAP directory server is more efficient if no additional fields are mapped to the LDAP server. This field is used for authentication only and does not affect any LDAP mappings that exist on the operator file for data warehousing purposes

Note: To use this field, LDAP connection information must be configured, either at the system level via the `scldapconfig` file; or in the LDAP mapping for the `operator` file via the scldapfile table. It is not necessary to map any fields in the `operator` file to LDAP.

Special considerations for Horizontally Scaled Service Manager

Service Manager can be installed horizontally scaled over several machines. To enable LDAP in this environment, ensure to have all LDAP parameters be the same in all involved `sm.ini` files, especially the ldapbinddn and ldapbindpass settings.

Troubleshooting your directory server integration

List of LDAP error codes

- **LDAP Error 3 Time Limit Exceeded** discussed in section The ldaptimelimit parameter
- **LDAP Error 4 Sizelimit Exceeded** discussed in section Queries that Return Incorrect Results
- **LDAP Error 15 No Such Attribute Exists** discussed in section Mismatched data on the Service Manager and directory server
- **LDAP Error 19 Constraint Violation** discussed in section Mismatched data on the Service Manager and directory server
- **LDAP Error 32 No such object found** discussed in section The DN is not located
- **LDAP Error 49 Incorrect Credentials** discussed in section The login credentials do not match
- **LDAP Error 50 Insufficient Access to Make Change** discussed in section Mismatched data on the Service Manager and directory server
- **LDAP Error 81 Can't Contact LDAP Directory Server** discussed in section Network connectivity issues
- **PRNG not seeded** discussed in section "PRNG not seeded" message received during certificate generation

A considerable amount of planning is involved when integrating Service Manager with a directory server. The interface is highly customizable, and configuration can be complicated. This chapter describes techniques that you can use to troubleshoot your integration in the following areas:

- Network connectivity problems
- Authentication problems
- Data retrieval and manipulation errors
- SSL configuration problems

Note: The first step for troubleshooting an LDAP issue is to enable the Service Manager LDAP logging capability using the ldapstats:1 parameter, which should be placed in the `sm.ini` file. Because the ldapstats parameter produces an abundance of log statements in the Service Manager log file, it should be used only when debugging an LDAP issue.

Network connectivity issues

The most common problem with connectivity is an incorrect LDAP server name in either the Service Manager LDAP mapping system level specifications or the LDAP File Mapping forms. The server name must be recognizable to the network, and the Service Manager server must have access to it. A common indicator of a connectivity problem is an "**LDAP Error 81 Can't Contact LDAP Directory Server**" message in your Service Manager log file.

The best way to verify your LDAP Server Name value is to use the operating system's `ping` command, a common method for troubleshooting device accessibility. The `ping` command uses a series of Internet Control Message Protocol (ICMP) echo messages to determine whether a remote host is active. The command sends an echo request packet to an address and then waits for a reply. The `ping` is successful only if the echo request arrives at the destination device, and only if that device can reply with an echo within a specified period called a *timeout*.

If the `ping` operation fails, verify that you have the correct name or IP address for your directory server. Your LDAP administrator can help with this.

If your LDAP Server Name is correct, ensure that you are using the correct LDAP port. Service Manager defaults to port 389 for non-SSL connections and port 636 for SSL connections. Check with your LDAP administrator to determine the correct value.

If the LDAP Server and port number are valid and you can still not connect to the directory server, there may be a conflict on the port. Multiple servers cannot use the same port for network communications. Check with your system administrator to resolve this issue.

Authentication problems

Authentication problems usually result from an incorrectly formatted DN in Service Manager. During the authentication process, Service Manager sends to the directory server a bind request, which includes the DN and the password of the user who is attempting to log in. If the DN is not located, or if the username and password credentials do not match, authentication is denied.

The DN is not located

An "**LDAP Error 32 No such object found**" message in the Service Manager log file indicates that the DN is not formatted correctly. An "Error 32 LDAP Object Not Found" message always occurs when a search has been sent to the directory server and the item cannot be located. It is usually caused by an incorrectly formatted DN or a search that is too restrictive. To resolve this problem:

- Contact your LDAP administrator to get the exact format of the bind DN, and check whether it contains the attribute that you have mapped to the primary key from the `operator` file. Try to uncheck the DN Contains Primary Key flag.
- Verify whether your directory server allows anonymous queries. If it does not, add the ldapbinddn and ldapbindpass parameters to your `sm.ini` file.
- Determine whether your directory server is case-sensitive. If it is, ensure that the username was entered in the correct case when you logged into Service Manager.
- Verify that your LDAP Base Directory is set at a level that will allow your DN to be located in the directory server.

Error 32 indicates that you need to recheck the format of the DN and the search criteria that were sent to the directory server. Error 49 indicates that the object was located in the directory server:

The login credentials do not match

An "**LDAP Error 49 Incorrect Credentials**" message occurs whenever the password does not match the password expected for the DN that is attempting to bind to the directory server.

If you receive an "Error 49 Invalid Credentials" message in the Service Manager log file, verify that you are entering the correct password during login.

If you do not receive the Error 49 but do get a message from Service Manager indicating that the login information is incorrect, an operator template is not set up for users who are in the directory server but do not have an operator record.

Example for an error situation:

```
RTE I LDAP: SASL BIND for OU=ETM,DC=INTRANET,DC=SAMPLE,DC=com,
sAMAccountName=doej returned 49 (Invalid credentials)
RTE I LDAP: callback function for referral rebind registered for DN
OU=ETM,DC=INTRANET,DC=SAMPLE,DC=com,sAMAccountName=doej LDAP server
INTRANET:389
RTE D |OpenLDAP| ldap_err2string
RTE I LDAP: Message from LDAP server: Invalid credentials, error code =
49
```

This error is caused by an incorrect DN order. The order within the DN is very important:

Wrong: OU=ETM,DC=INTRANET,DC=SAMPLE,DC=com,sAMAccountName=doej

Correct: sAMAccountName=doej,OU=ETM,DC=INTRANET,DC=SAMPLE,DC=com

Data retrieval and manipulation errors

Retrieving and manipulating directory server data from Service Manager can be frustrating if the interface is not configured correctly. Appropriate access rights on the directory server and valid settings such as the LDAP Base Directory in Service Manager must be in place. Common problems are:

- Slow or inefficient queries
- Queries that return incorrect results
- Queries that do not display the expected number of records
- Mismatched data between Service Manager and the directory server

Slow or inefficient queries

If you notice that retrieving data from an LDAP mapped file is taking too long to process, search the Service Manager log file for the query that Service Manager is sending to the directory server, which starts with the line "LDAP Query". After you determine that the query is formatted correctly, review it with your LDAP administrator and consider either adding a search filter or changing the base directory.

If you are accessing a form in Service Manager that contains a field or link to an LDAP mapped file, a query is issued to the directory server. This can be misleading because some users do not realize that the directory server query is processing in the background, which can delay form processing.

For example, assume that an out-of-the-box implementation of Service Manager where the Add/Edit Contact button on the cc.incquick form takes more than two minutes to return a result if the `contacts` file is mapped to a directory server. The default query used by this form starts with the crosshatch character (#). These types of queries are extremely inefficient against a directory server. To improve performance, follow these steps to change the query to an exact match query, which begins with the equal sign (=):

1. Enter **link** at the Service Manager command prompt to open the Link Manager.

2. In Link Manager, open the cc.incquick link record.
3. Search for $contact in the Source Field Name column.
4. In the same row, find the query in the AddQuery column.
5. Change the crosshatch character (#) to an equal sign (=).
6. Save the link record.

If the query cannot be modified, you can use the ldapmaxrecord and the ldaptimelimit parameters to set limits on the query. Both ensure that Service Manager performs within the limitations you specify when retrieving data from the directory server.

Another way to improve query performance is to limit the number of Service Manager fields that you map to the directory server. Mapping unnecessary fields not only causes Service Manager to execute unnecessary queries, it can also cause erroneous data to be returned. For example, do not map a field to an attribute that you know always contains blank or useless data.

Queries that Return Incorrect Results

If your directory server queries are not returning the data you expect, first search the Service Manager log file for the query that Service Manager is sending to the directory server, which starts with either "LDAP Query" or "LDAP: Query." If the query is formatted correctly, try any of the following to resolve the issue.

- Check with your LDAP administrator and determine whether there is a record or time limit set on the directory server. Limitations can alter the number of records you receive. If you encounter one of these limitations, you will receive an "**LDAP Error 4 Sizelimit Exceeded**" message in the Service Manager log file.
- If the query is the result of displaying a form, verify that the CaseConversion property of the query search fields is set to **none** in the Forms Designer. Otherwise, the query may not find case-sensitive records.
- Ensure that the LDAP Base Directory is set to a level at or above the data you wish to query. If your setting is too limited, you may not retrieve all your results.
- When performing data warehousing without LDAP authentication, ensure that the ldapbinddn account is valid and has appropriate access rights to the data for which you are searching.

Queries that do not display the expected number of records

If your query does not return the number of records that you are expecting, check whether a time or record limitation was breached on the directory server. This usually happens when Service Manager does not have LDAP set to the primary data source. When issuing this type of query, Service Manager queries its database for matching records before it issues a true search to the directory server. If the directory server reaches the maximum limitation, it returns only a portion of its results. Each record is then matched with its corresponding record in Service Manager. If no corresponding record exists, the record is not displayed. Therefore, displayed results do not necessarily match the maximum number of records returned by the directory server. The only way to resolve this issue is to remove the time and record limitations from the directory server. Discuss this option with your directory server administrator.

Mismatched data on the Service Manager and directory server

If you notice inconsistency between the data in Service Manager and the data on your directory server, make sure there are no error messages that indicate add or delete operation failures in your Service Manager log file. These messages can be found by searching for "LDAP Add" or "LDAP Delete" in the log file. Such messages have the entire DN for the object you are adding or deleting. After you have verified that the DN is set correctly, try any of the following actions to resolve the issue:

- Verify that the bind user has appropriate access rights on the directory server. If not, an "**LDAP Error 50 Insufficient Access to Make Change**" error message appears in the Service Manager log file.

For insertions only

- Ensure that all required attributes are mapped, or included in the LDAP Base Attr Field. If they are not, an "**LDAP Error 19 Constraint Violation**" error message appears in the log file.
- Check the attributes that are mapped to the Service Manager file. If they do not exist on the directory server, an "**LDAP Error 15 No Such Attribute Exists**" error message appears in the log file.
- If you are using the LDAP DN Template for Inserts field, check the format and make sure that the attributes and Service Manager fields are correct.

SSL configuration issues

Due to its complexity, you may encounter several different problems when configuring your directory server interface to communicate with SSL. Because SSL is a standard protocol, your best sources of information are found by searching the Internet or reading SSL documentation. Detailed information about OpenLDAP or OpenSSL errors is available at www.openldap.org or www.openssl.org. Two of the most common SSL errors are described below.

Certificate is generated with an incorrect server name

When generating an SSL certificate, you provide the directory server name, which must match the LDAP Server field in Service Manager. That means the directory server name specified in the server certificate must be pingable. If not, the directory server to which Service Manager requests a connection does not match the one listed in the certificate; verification fails; and the connection is denied. Consult your directory server administrator to determine the exact syntax of the fully qualified domain name.

"PRNG not seeded" message received during certificate generation

To communicate securely using encryption, an SSL connection negotiates the encryption based on random data. Some platforms offer special devices to return random data. With operating systems that do not (including AIX®, HPUX, Solaris™) OpenSSL does not receive random data and cannot complete the request. This error can be resolved by executing the following command:

openssl rand –rand <input file> -out ~/.rnd 1024

This command creates a file called `.rnd` in the user's home directory with the length of 1024 bytes. This file is used by other OpenSSL commands as the seed for random data. With every `openssl` command, the file is updated. The input file can be any file, and the more random or unique it is the better. One possibility is to use the `sm.log` file as the input file because it contains a lot of unpredictable data. The size of the input file must be at least 1024 bytes.

Appendix A - Acronyms and Abbreviations

Term	Definition
LDAP	Acronym for *Lightweight Directory Access Protocol*, a set of protocols used for accessing information directories
DN	Acronym for *Distinguished Name*, a fully qualified name that designates each item in a directory structure with an individual key name.
DIT	Acronym for *Directory Information Tree*, the complete collection of directory objects logically represented in a hierarchical tree
SSL	Acronym for *Secure Sockets Layer*, a protocol developed by Netscape® Communications Corporation for transmitting private documents via the Internet
Base DN	The distinguished name (DN) that identifies the starting point of a search
Root DSE	An entry that is located at the root of the DIT
Bind User	The directory server account to which Service Manager binds
PEM	Acronym for *Privacy Enhanced Mail*, which is a widely used certificate format for creating secure network connections

Appendix B - References

- "Understanding Directory Services," December 2001, SAMS Publishing
- "LDAP System Administration," March 2003, O'Reilly & Associates
- OpenLDAP® home page – www.openldap.org
- OpenSSL home page: – www.openssl.org
- "Implementing LDAP,"", July 2000, Wrox Press

For more information

Please visit the HP Software support Web site at:

www.hp.com/go/hpsoftwaresupport

This Web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Note: Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following URL:

www.hp.com/go/hpsoftwaresupport/new_access_levels

To register for an HP Passport ID, go to the following URL:

www.hp.com/go/hpsoftwaresupport/passport-registration