Implementing the Service Location Protocol

The Service Location Protocol (SLP) is an Internet standard protocol (RFC 2165) that enables client applications to dynamically discover services in TCP/IP networks. Novell® provides implementations of SLP for NetWare®, Windows 95, Windows 98, Windows NT, and Windows 2000.

Understanding SLP Components

SLP defines three types of agents:

- User Agents
- Service Agents
- Directory Agents

Functionality of SLP Directory Agents is not provided for Linux or Solaris systems.

User Agents

User Agents work in behalf of client applications to retrieve service URLs and attributes of desired network services. Client applications can request all URLs of a specific service type or narrow the search by requesting only services of a certain type with specific attributes.

If no Directory Agents are available to the User Agent, the SLP request is sent to multiple services (multicast) using the Service Location General Multicast Address (224.0.1.22, see RFC 2165). All Service Agents holding service information that satisfy the request unicast the reply (using UDP or TCP) directly to the requesting User Agent.
If a Service Agent has the requested service information, it replies. If multiple Service Agents reply, the User Agent combines the replies before presenting them to the client application. If a Directory Agent is available, the User Agent unicasts the SLP request to the Directory Agent rather than sending a multicast request. The Directory Agent always unicasts a reply even if the answer indicates that no services are available.

User Agents send the following SLP requests:

<table>
<thead>
<tr>
<th>Table 64</th>
<th>SLP Requests Sent by User Agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>Description</td>
</tr>
<tr>
<td>Service Type Request</td>
<td>Returns all active service types.</td>
</tr>
<tr>
<td>Service Request</td>
<td>Returns the service URLs of a specific type.</td>
</tr>
<tr>
<td>Attribute Request</td>
<td>Returns the attributes of a specific service URL.</td>
</tr>
</tbody>
</table>

User Agents process the following SLP replies:

<table>
<thead>
<tr>
<th>Table 65</th>
<th>SLP Replies Processes by User Agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply</td>
<td>Description</td>
</tr>
<tr>
<td>Service Type Reply</td>
<td>Contains the list of known service types.</td>
</tr>
<tr>
<td>Service Reply</td>
<td>Contains a list of the requested service URLs.</td>
</tr>
<tr>
<td>Attribute Reply</td>
<td>Contains the requested attributes of a specific service URL.</td>
</tr>
<tr>
<td>DA Advert</td>
<td>Sent by Directory Agents to indicate their existence.</td>
</tr>
</tbody>
</table>

Novell provides implementations of User Agents for NetWare, Windows 95/98, Windows NT, and Windows 2000.

**Service Agents**

Service Agents (defined by RFC 2609 (http://www.openslp.org/doc/rfc/rfc2609.txt)) work in behalf of network service applications to passively advertise service URLs representing the services provided. Network service applications register the service URL and attributes that define their network service with the Service Agent.
The Service Agent maintains a local database of registered service information. The Service Agent does not broadcast or multicast the registered services on the network but passively waits for SLP requests to be multicast from User Agents.

If Directory Agents are present, the Service Agent registers the services with each Directory Agent. Service Agents send the following SLP requests:

**Table 66 SLP Requests Sent by Server Agents**

<table>
<thead>
<tr>
<th>Request</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Registration</td>
<td>Registers a service URL and its attributes with a Directory Agent.</td>
</tr>
<tr>
<td>Service Deregistration</td>
<td>Deregisters a service URL and its attributes from a Directory Agent.</td>
</tr>
<tr>
<td>Attribute Request</td>
<td>Returns the attributes of a specific service URL.</td>
</tr>
</tbody>
</table>

Service Agents process the following SLP requests:

**Table 67 SLP Requests Sent by Server Agents**

<table>
<thead>
<tr>
<th>Request</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Type Request</td>
<td>Returns all held service types.</td>
</tr>
<tr>
<td>Service Request</td>
<td>Returns the service URLs of a specific type.</td>
</tr>
<tr>
<td>Attribute Request</td>
<td>Returns the attributes of a specific service URL.</td>
</tr>
<tr>
<td>DA Advert</td>
<td>Sent by Directory Agents to indicate their existence.</td>
</tr>
</tbody>
</table>

Novell provides implementations of Service Agents for NetWare, Windows 95/98, Windows NT, and Windows 2000.
Directory Agents

The Directory Agent maintains a database of service URLs representing network services. Service Agents acting in behalf of network applications register service URLs with the Directory Agent.

Multiple Directory Agents can be deployed in a network. Service Agents register their service URLs with each known Directory Agent, maintaining consistent service information among all Directory Agents.

RFC 2165 does not define a protocol for synchronizing service information between Directory Agents. To compensate, Novell SLP Directory Agents support a feature known as Directory mode.

Directory Agents configured for Directory mode use Novell eDirectory as a common, distributed, replicated data store through which multiple Directory Agents can share service URLs. This enables Directory Agents to report service URLs that were registered with other Directory Agents, configured in Directory mode, as well as report the services registered by local Service Agents.

Such reporting reduces network traffic by eliminating the need for Service Agents to register with every Directory Agent in the network. This reduction is particularly advantageous for large enterprise networks with WAN backbones.

Novell provides implementations of Directory Agents for NetWare, Windows NT, and Windows 2000. Directory Agents running on NetWare operate only in Directory mode. Directory Agents running on Windows NT or Windows 2000 can operate in Directory mode or Local mode. A Directory Agent operating in Local mode does not share service information with other Directory Agents; it operates autonomously, as defined by RFC 2165.

The Directory Agent is responsible for processing the following SLP protocol messages:

- Service Registration
- Service Deregistration
- Service Type Request
- Service Request
- Attribute Request
- Directory Agent Advertisements
These SLP messages enter, delete, or query for service URLs and associated attributes in the Directory Agent’s service database.

For more information on these message types, refer to RFC 2165.

**Service Registration**

To register service URLs and their attributes with Directory Agents, Service Agents send Service Registrations. Each service URL includes a lifetime which, if it expires, causes the Directory Agent to delete the service from its database.

The Service Agent must refresh the service registration at least once during the service’s lifetime. The service lifetime ensures that the Directory Agent can eventually purge its service cache of service URLs registered by Service Agents that do not deregister their service URLs.

**Service Deregistration**

To remove a service URL and its attributes from the Directory Agent service cache, Service Agents send Service Deregistrations to Directory Agents. This action can occur if the network application is terminating or if the Service Agent is being shut down.

**Service Type Request**

To obtain a list of active service types on the network, User Agents send Service Type Requests to Service Agents (multicast) or Directory Agents (unicast). Service Agents and Directory Agents return their known service types with a Service Type Reply, which is unicast to the requesting User Agent.

**Service Request**

Service Requests are sent by User Agents to Service Agents (multicast) or Directory Agents (unicast) in search of service URLs representing desired services. Service URLs matching the request criteria are returned in a Service Reply, which is unicast to the requesting User Agent.

Service Requests can be general in nature and request all URLs of a specific service type, or they can contain a predicate that specifies that only services of a certain type with specific attributes be returned.
Attribute Request

To retrieve one or more attributes of a specific service URL, User Agents send Attribute Requests to Service Agents (multicast) or Directory Agents (unicast).

The Attribute Request can be general, requesting that all attributes be returned. Also, the Attribute request can contain an Attribute Select list, identifying one or more specific attributes to be returned.

The requested attributes are returned in an Attribute Reply that is unicast to the requesting User Agent.

Directory Agent Advertisement


Directory Agent Advertisements contain:

- The service URL for the Directory Agent.
- Other configuration information that help User Agents and Service Agents determine which Directory Agents to direct SLP requests.

If multicasts are not enabled or allowed in a network, User Agents and Service Agents can be configured with the network addresses of Directory Agents. In such a case, the User Agent and Service Agent query (with a Service Request of type directory-agent) the Directory Agent for its Directory Agent Advertisement.

For a complete description of User Agent, Service Agent, and Directory Agent synchronization, see RFC 2165.

SLP Scopes

An SLP scope is a defined group of network services. Scopes enable one or more groups of users to easily use network services.

To define a scope, you can use criteria that help you organize and administer network services. If you have configured users to use a specific set of scopes, you can effectively assign a set of available services to those users.
You can create scopes to reflect departments in your company. For example:

- A Human Resources scope groups the services unique to the Human Resources department
- An Accounting scope groups the resources pertaining to the Accounting department

With these scopes, you can configure users in the Human Resources department to use the Human Resources scope. Also, you can configure users in the Accounting department to use the Accounting scope. Users requiring services in both departments can be configured to use both scopes.

Likewise, services can be grouped according to geographical location. You can define an SLP scope for each city or country where your company has an office. You can configure users in each locality to use the scope defined for their office. If a user needs access to services in multiple sites, you can configure that user to use the scopes of all necessary sites.

In addition to dividing services according to organizational and geographical criteria, you can define scopes to hold common services that multiple groups must share. This feature allows users to locate shared services while keeping their unique services locally.

Another reason to use scopes is to enhance the scalability and performance of SLP. Service registrations are organized and stored according to the scope in which they have been registered. Directory Agents are configured to service one or more scopes. If all services in a network are contained in a single scope, and, therefore, a single service cache, the amount of service information can become unwieldy and difficult to manage. Response times might suffer because of the immense amount of data that must be searched to satisfy a request.

Therefore, in large network environments, it is better to group the services into scopes and then assign one or more Directory Agents to service the scopes applicable to the users that will be utilizing the Directory Agent.

Service Location Protocol 1 (RFC 2165) defines the default operating configuration of User Agents, Service Agents, and Directory Agents to be unscoped, meaning that no scopes are configured. This means that all services are maintained as if in a single scope that has no name.

Additionally, special rules apply when registering with, or requesting services from, unscoped agents. In particular, all services regardless of scope should be registered with unscoped Directory Agents. But if an unscoped request is made to an unscoped agent, only those services registered as unscoped can be
returned. On the other hand, a scoped request will return all services from the
requested scope as well as all unscoped services matching the request criteria.

When both scoped and unscoped agents are used in the same network, results
are often confusing and sometimes inconsistent. Therefore, Service Location
Protocol 2 (RFC 2608) removed unscoped operations from the Service
Location Protocol and redefined the default operating configuration to use a
default scope named Default.

To eliminate the confusion associated with mixing unscoped and scope agents
in a single network and to facilitate eventual migration to SLP 2, we
recommend that users always configure SLP to use scopes.

For the following reasons, generally use scopes to organize SLP service:
- Services are registered into and retrieved from a scope
- Many SLP configuration parameters are set according to scopes
- Directory Agents are configured to service one or more scopes
- User Agents and Service Agents determine which Directory Agent to
  query based on the scopes the Directory Agent is supporting.

Fundamental to the successful organization, deployment, and administration
of SLP in a network, scopes are a valuable tool in controlling the availability
of services in the network.

How SLP Works

User Agents and Service Agents interact on behalf of client applications and
network services to dynamically locate network services. The following
interactions are discussed in this section:
- “SLP with a User Agent, Service Agent, and No Directory Agent” on
  page 478
- “SLP with a User Agent, Service Agent, and Directory Agent” on page
  479

SLP with a User Agent, Service Agent, and No Directory Agent

Figure 33 illustrates how Service Agents and User Agents interact without a
Directory Agent in the network. When a network application is started, it
registers its service URL and attributes with the Service Agent. The Service
Agent stores a copy of the service information in its local service cache. The
Service Agent remains silent, meaning that the service is not multicast or broadcast on the network.

**Figure 33  SLP User Agent and Service Agent Interaction**

When a client application queries the User Agent for a network service, the User Agent in search of service information multicasts a Service Request. The Service Agent receives the Service Request and consults its local service cache to see if it holds a service matching the criteria of the Service Request. If so, the Service Agent containing the requested service information unicasts a Service Reply to the User Agent.

If multiple Service Agents reply, the User Agent combines the results before returning them to the client application. This same scenario occurs for Service Type and Attribute Requests. When the network service is terminated, it deregisters its service with the Service Agent, which deletes the service from its local service cache. The Service Agent remains silent.

**SLP with a User Agent, Service Agent, and Directory Agent**

Figure 34 on page 480 illustrates how Service Agents and User Agents interact with Directory Agents to advertise and locate network services. When a network application is started, it registers its service URL and attributes with the Service Agent. The Service Agent maintains its own copy of the service information and then unicasts a Service Registration (that has the new service
information) to the Directory Agent. The Directory Agent saves the service information in its local service cache.

Figure 34  SLP Directory Agent interaction

When a client application queries the User Agent for a network service, the User Agent in search of service information unicasts a Service Request to the Directory Agent. The Directory Agent returns a Service Reply that has the requested service URLs or an indication that no requested services are available. The same scenario is repeated by the User Agent and Directory Agent for Service Type Requests and Attributes Requests.

When the network service is terminated, it deregisters its service with the Service Agent, which deletes the service from its local service cache and then sends a Service Deregister request to the Directory Agent. The Directory Agent then deletes the indicated service from its service cache.

Understanding Local Mode

Novell Directory Agents can be installed and configured so that the Local mode operation can do the following:

- Provide a centralized repository of service URLs.
- Facilitate the use of SLP scopes.
Create customized scopes by selectively gathering services from other scopes.

Proxy scopes directly supported by other Directory Agents or Service Agents.

Improve SLP scalability, performance, and network efficiency.

Facilitate the use of SLP in networks not supporting IP multicast.

Act as private Directory Agents for closed groups of Service Agents and User Agents through Private Mode.

Filter service content of SLP scopes based on service type, service URL, service lifetime, and the IP address of the Service Agent or User Agent.

Central Repository

Directory Agents function as a centralized data store for service URLs that are registered by Service Agents and solicited by User Agents. Because Directory Agents hold all the services for each configured scope, User Agents can obtain all desired service information with a single request and reply. By contrast, in networks without Directory Agents, User Agents issue a multicast request and might receive many replies.

SLP Scopes

Directory Agents are configured to support one or more SLP scopes. (An unscoped operation is similar to supporting a single scope.) Directory Agents collect and store service URLs and their associated attributes according to the scope in which the services are registered. Service Agents and User Agents obtain the scopes supported by a Directory Agent from a Directory Agent’s DA Advert message. In this way, User Agents and Service Agents can dynamically detect and utilize the scopes configured for each Directory Agent. In networks without Directory Agents, Service Agents and User Agents must be configured with the SLP scopes they will use.

Customized Scopes

Novell Directory Agents allow the network administrator to create customized scopes by pulling service information from one scope and storing it in a different scope. This is a variation of the scope proxy feature because the custom scope name is different than the scope being proxied.
For example, if a network administrator wants to create a custom scope for a single group of users containing only specific service URLs and attributes, the custom scope is configured on the local Directory Agent and the address of the scope authority servicing a target scope and the target scope’s name is configured as a proxy address for the custom scope. The content of the custom scope can be further controlled by adding filters that apply only to the custom scope.

When the services are retrieved from the scope authority and registered in the custom scope, the attributes of the service are modified to indicate that the service is now part of the custom scope. The group of users can then be configured to use only the custom scope with the network administrator controlling the service information available to them. Using this same technique, a hierarchy of scopes can be created to reflect the administrative groupings of services that best fit your network user’s needs.

**Proxy Scopes**

Novell Directory Agents can be configured to proxy scopes supported natively by other Directory Agents, also referred to as scope authorities. Instead of having every Service Agent register with every Directory Agent in the network, Service Agents can be configured to register with a single or small subset of Directory Agents. The other Directory Agents in the network are then configured to proxy the scopes of the central Directory Agents, which act as the authorities for the proxied scopes.

When a Directory Agent is configured to proxy a scope supported by another Directory Agent, the proxy agent downloads the scope information at configured intervals and then acts as a local service cache for that scope. This can be advantageous for remote sites reachable over WAN segments. Rather than having User Agents in remote sites interacting with Directory Agents over the WAN, a proxy Directory Agent can be deployed in the remote site, keeping all SLP service queries within the local site’s network.

**Scalability and Performance**

Because service information can be registered and obtained with a single unicast request and reply, the operation of SLP becomes more efficient and hence more scalable. Because each interaction with a Directory Agent always results in a reply, the time required to resolve a service request is kept to a minimum. When a User Agent issues a multicast request, it must wait a period of time before determining if all answers have been received. This is because
Service Agents and Directory Agents do not respond unless they can answer the query. As a result, the User Agent must pause while waiting for replies, estimating when all possible answers have been received. But as soon as a User Agent receives a reply from a Directory Agent it can process the response immediately.

All protocol interactions with a Directory Agent are performed using unicast messages. If multicast is not supported on your network, deploying a Directory Agent and configuring the Service Agents and User Agents with the IP address of the Directory Agent (through local configuration or DHCP) allows SLP to be used in networks that do not support multicast addressing.

Private Mode

In addition to the features listed above that are defined by the SLP protocol, Novell Directory Agents support other value-added features that assist the network administrator in deploying SLP within their network. Novell Directory Agents can be configured to operate in Private mode. When configured for Private mode, the Directory Agent does not multicast Directory Agent Advert messages or answer multicast requests, thus making the Directory Agent undiscoverable by dynamic means. To use a Directory Agent configured in Private mode, User Agents and Service Agents must be configured with the address of the private Directory Agent.

This allows the network administrator to create closed groups of users of one or more private Directory Agents. Private Directory Agents are also a valuable tool in piloting new versions of the Directory Agent or testing new configurations without disturbing the operating network.

Filtering

When a Directory Agent is operating in Local mode, network administrators can configure filters that control which service URLs are accepted for registration and which service URLs are returned in service replies. The filters are configured on a per-scope basis, allowing network administrators to customize the content of each scope separately. The filtering criteria include service type, specific URLs, service lifetime, and the address of the Service Agent or User Agent making a request. One or more filter criteria can be specified for each filter.
Understanding Directory Mode

Novell Directory Agents can be configured for operation with eDirectory to

- Provide a single point of configuration and administration of SLP agents.
- Share service information among multiple Directory Agents.
- Conserve network bandwidth.
- Perform all operations supported by Local mode.

SLP Directory Agents, scopes, and services can be configured and managed through eDirectory. This provides a single point of control to network administrators implementing and managing SLP in their networks.

Directory Agents are configured using Directory Agent objects containing configuration information for the Directory Agent. SLP Scope container objects can be configured to represent SLP scopes. A Directory Agent object contains fully distinguished names of one or more SLP Scope container objects which indicate the scopes the Directory Agent is to service. Services registered with the Directory Agent are stored in the SLP Scope container object as SLP Service objects. Each SLP Service object includes the service’s service URL and attributes. SLP Service objects can be manipulated just like any other eDirectory object, including deleting and copying to another SLP Scope container object.

Novell Directory Agents can share service information by using eDirectory as a common data store for service URLs and their attributes. In this manner, the distributed, replicated, and synchronized nature of information stored in eDirectory is leveraged to eliminate the need for every Service Agent in the network to directly communicate with every Directory Agent in the network. SLP Scope container objects representing SLP scopes are configured in eDirectory. Directory Agents, configured to service the scope, cache each registered service locally and store each service and its attributes as an SLP Service object in the SLP Scope container object. These Directory Agents also populate their local service cache with services obtained from the SLP Scope container object. By storing and retrieving from the shared SLP Scope container objects, Directory Agents can return service URLs and attributes for services registered by remote Service Agents.

Because Directory mode facilitates the sharing of service information through common SLP Scope container objects, Service Agents are not required to register a service with every Directory Agent in order for their services to be known throughout the network. This reduces configuration complexity and
reduces network traffic. Using this capability, Service Agent to Directory Agent interactions can be isolated to local segments within the network similar to User Agent-to-Directory Agent interactions.

**How SLP Works in Directory Mode**

Novell Client™ software uses the User Agent to go to an SLP Directory Agent or into eDirectory to reach out to other LAN or WAN segments, as shown in Figure 35.

This method does not rely on service information obtained from routers. Instead, eDirectory is used for global communication of information. Through this method, service updates on local segments are just as reliable and dynamic as on IPX™ SAP-based networks.

**Figure 35  Integrated Network Services Discovery**
SLP eDirectory Objects

Using ConsoleOne® you can manage the following eDirectory objects used by SLP:

- “SLP Scope Container Object” on page 486
- “SLP Service Object” on page 487
- “Directory Agent Object” on page 487
- “Server Object” on page 487

The SLP Scope container object represents an SLP scope and stores SLP Service objects.

SLP Service objects represent a network service discovered through the Service Location Protocol. They contain all of the SLP information about the network service, including its network address and attributes.

The SLP Directory Agent object represents an SLP Directory agent.

SLP Scope Container Object

SLP uses the SLP Scope container object, which defines a logical grouping of services. The Scope object allows network administrators to logically group services according to geographical, geopolitical, service type, or any other administrative criteria in order to control distribution or visibility on the network. The primary goal of the Scope object is to enhance the scalability of gathering and distributing network service information.

The SLP Scope object is the storage container for SLP service information. Each object contains all the SLP Service objects for the specific scope. The eDirectory administrator can replicate the container into other partitions within the tree or within federated trees. The object is a standalone entity within the eDirectory tree, and there is no relationship between its distinguished name, the tree name, and the scope name. When a Service Agent forwards a service record to a Directory Agent within a specific scope, the scope name is mapped to the Scope object by using the name attribute within the container object. The SLP Scope object must contain rights to read, write, and browse the container because the access rights of the Directory Agent object access are equivalent to the access rights of the Scope object. Because the Scope object uses distinguished name syntax, the Scope object can be moved to a different location in the tree. eDirectory will automatically change all values to reflect the new location.
SLP Service Object

The SLP Service object is a leaf object that represents a service registration. SLP Service objects are subordinate to the SLP Scope object and contain all information supplied by a service registration. SLP Service objects are stored in the appropriate SLP Scope object according to their scope.

Directory Agent Object

The SLP Directory Agent object is a leaf object that represents a single instance of a Directory Agent. Multiple Directory Agents cannot share a single object. This object defines the Directory Agent’s configuration, scope, and security. The Directory Agent uses this object to log in to the server and operate under the access control requirements assigned to the Server object.

Server Object

The NetWare installation program creates an NCP_SERVER object for every server within the tree. The Directory agent adds an attribute to the NCP_SERVER class definition called SLP Directory Agent DN. The SLP Directory Agent DN contains the distinguished name of the Directory Agent object. It is used as a pointer from the Server object to the Directory Agent object.

Novell’s Implementation of SLP

The following sections discuss Novell’s implementation of the Service Location Protocol (SLP) specification.

- “Novell’s User Agents and Service Agents” on page 487
- “The Novell Directory Agent” on page 494

Novell’s User Agents and Service Agents

The Novell Client includes software for User Agents and Service Agents. The software is installed automatically during a client installation when one of the IP protocol options is chosen.

SLP must be available for the client to function and should be used before other Service Name resolving methods (eDirectory, SAP, etc.) by the client. Otherwise, changing most of the SLP configuration parameters will have no
functional effect on a Workstation/UA because SLA is either not available or is not being used to resolve service names.

To configure the parameters, go to the Novell Client Configuration property pages (right-click Network Neighborhood or My Network Places, then click Properties > Services > Novell Client for Windows NT > Properties).

**SLP Configuration Parameters**

**Service Location Tab**

The following paragraphs describe the options found on the Service Location tab of the Novell Client for Windows NT.

**Scope List:** Defines what SLP scopes the UA will participate in. Controls what DAs and SAs the UA will communicate with for SLP Service queries.

If the SA/DA is not in a scope specified at the UA, the UA will not send a request or accept a response from it. The exception to this is if there is no scope specified, then the UA will participate in the unscoped scope.

Scope entries can be set in a precedence order by using the Up- and Down-arrows. Scopes can come from three different sources: Static, DHCP, and Dynamic. As with other SLP settings, Static scopes have a higher preference than DHCP scopes, and DHCP scopes have a higher preference than Dynamic scopes for locating services.

**Table 68 Scope List Values**

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>Any entry will be accepted, but should match the scope name used with the SAs/DAs you want to communicate with.</td>
</tr>
</tbody>
</table>

**Static:** Checking the Static check box will prevent the client from dynamically adding scopes discovered from the known active DAs. The active DAs can be checked by executing the SLPINFO command. If the Static check box is not checked, then when the client discovers a DA that participates in a scope that was previously unknown to the client, the client will add the scope to its list in memory and can then query for SLP Services in that scope.
Table 69 Static Values

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Checked/Unchecked (On/Off)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td></td>
</tr>
</tbody>
</table>

**Directory Agent List:** This parameter controls what DAs a client is statically configured to communicate with. This is not necessarily a complete list of the DAs the client is aware of. You must use SLPINFO with the /D command to be sure of what DAs the client has discovered and their status (Active/Inactive).

Table 70 Directory Agent List Values

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>Any IP address or DNS-resolvable host name for a NetWare server running SLPDA.NLM</td>
</tr>
</tbody>
</table>

**Static:** Checking this check box will prevent dynamic DA discovery and use only DAs discovered using the Static or DHCP method.

- The UA will not send an initial DA multicast requesting a response from all DAs that can hear the request.
- Any DA that multicasts a DA Advertisement (DA_ADVERT) will be ignored. Normally the UA will add any DA that does a DA Advertisement. DAs advertise when they are first loaded and also periodically based on the Heartbeat parameter.

Table 71 Static Values

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Unmarked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>Marked/Unmarked (On/Off)</td>
</tr>
</tbody>
</table>

**Active Discovery:** Unchecking this check box requires that the UA contact a DA for an SLP Request. The UA will not multicast the request to SAs. The combination of Static enabled and Active Discovery disabled will entirely prevent the UA from multicasting. When this setting is disabled, you are required to put at least one entry in the Directory Agent List; otherwise, the UA has no method for querying for SLP services.
Advanced Settings Tab

The following paragraphs describe the options found on the Service Location tab of the Novell Client for Windows NT.

Give Up on Requests to SAs: Timeout (in seconds) for an SLP Request to an SA. This parameter is not used to time out requests to DAs because there is a separate setting for that.

Table 72 Active Discovery Values

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Checked (On)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>Checked/Unchecked (On/Off)</td>
</tr>
</tbody>
</table>

Table 73 Give Up on Requests to SAs Values

<table>
<thead>
<tr>
<th>Default Value</th>
<th>15 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>1 to 60,000 seconds (16.67 hours)</td>
</tr>
</tbody>
</table>

SLP Cache Replies: Every time the UA receives an SLP Service reply from a DA/SA, it will be cached/saved at the UA for the amount of time specified in the SLP Cache Replies parameter. When SLP receives a request it will first check its cache before generating a network packet to a DA/SA. If the cached information can be used to answer the request, it will be.

It is not recommended to set the time higher than one minute under normal SLP operations for the following reasons:

- During normal SLP communication, duplicate requests should occur within one minute of the original request, making longer caching unnecessary.
- The higher the setting, the more memory will potentially be required to cache this information.

Table 74 SLP Cache Replies Values

<table>
<thead>
<tr>
<th>Default Value</th>
<th>1 minute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>1 to 60 minutes</td>
</tr>
</tbody>
</table>
**SLP Default Registration Lifetime**: This parameter determines the registration lifetime of an SLP Service when an SA registers an SLP Service to a DA. The Novell Client not only includes the UA capabilities, but also the SA capabilities (the same as a server), so it is possible for a client workstation to be registering SLP services with a DA. However, it is unusual for a client workstation to be registering an SLP Service as an SA. Developers can write applications that register SLP Services from a client workstation using the WINSOCK 2 interface.

Examples of cases where a client workstation would register an SLP service include the following:

- An NT Domain Controller running NDS4NT and a local eDirectory replica.
- A Client workstation running the Compatibility Mode client (CMD) where the workstation is advertising an SAP (for example, 0x0640). CMD will convert the SAP to SLP and register it with any DAs the client has discovered.
- A third-party application is using WINSOCK to intentionally register an SLP Service.

When the Registration Lifetime of an SLP Service expires, the DAs it is registered with will remove this entry from its database. This is also used to determine when an SA (on a workstation or a server) needs to reregister a service with its DAs.

<table>
<thead>
<tr>
<th>Table 75</th>
<th>SLP Default Registration Lifetime Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>10,800 seconds</td>
</tr>
<tr>
<td>Valid Values</td>
<td>60 - 60,000 seconds</td>
</tr>
</tbody>
</table>

**SLP Maximum Transmission Unit**: Exactly the same as the TCP/IP MTU, which is the maximum size that an SLP packet can have. This setting is used to restrict the size of an SLP packet so that it does not exceed the capability of the infrastructure and does prevent resource-intensive packet fragmenting and reassembly.

<table>
<thead>
<tr>
<th>Table 76</th>
<th>SLP Maximum Transmission Unit Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>1,400 bytes</td>
</tr>
<tr>
<td>Valid Values</td>
<td>576 to 4,096 bytes</td>
</tr>
</tbody>
</table>
**SLP Multicast Radius**: This parameter specifies the maximum number of subnets (number of routers plus 1) that SLP multicasts can travel across. A value of 1 prevents multicasting from crossing any router. This is implemented in the Time To Live (TTL) setting of the UDP/TCP packet.

TTL is decremented by one of two conditions:

- The packet crosses a router
- The packet is buffered in a router for more than 1 second

<table>
<thead>
<tr>
<th>Table 77</th>
<th>SLP Multicast Radius Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>32 hops</td>
</tr>
<tr>
<td>Valid Values</td>
<td>1 - 32 hops</td>
</tr>
</tbody>
</table>

**Use Broadcast for SLP Multicast**: This parameter forces the SLP UA to use broadcast (all bits turned on in the Host ID portion of the address) where it would have normally used multicast.

This has the following different behaviors from multicast:

- Broadcast will not cross a router, so this limits the packet to the originating subnet.
- It might cause additional bandwidth usage because the packet will now need to be repeated from every switch port. Some switches are capable of tracking multicast registrations and that forward a multicast packet only from the switch ports that are registered for that multicast address.

<table>
<thead>
<tr>
<th>Table 78</th>
<th>Use Broadcast for SLP Multicast Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>Off</td>
</tr>
<tr>
<td>Valid Values</td>
<td>On/Off</td>
</tr>
</tbody>
</table>

**Use DHCP for SLP**: This parameter determines whether the SLP UA will attempt to locate a DHCP server that can provide SLP Scope and DA configuration information. Even if the workstation’s IP address is statically configured, SLP can still receive an SLP Scope and DA configuration from a DHCP server. The DHCP requests for SLP information are sent only as part of the SLP UA/SA initialization. SLP information is requested using the DHCP INFORM request and is sent in addition to the initial BOOTP Request.
(if the client is configured to obtain its IP address via DHCP/BOOTP). All SLP DHCP response information is combined, then SLP contacts each DA that has been configured by DHCP to determine the scopes supported by each DA.

Administrators who plan to never use DHCP to administer SLP information should set this parameter to Off to reduce the minimal traffic the broadcast for a DHCP server will require.

<table>
<thead>
<tr>
<th>Table 79</th>
<th>Use DHCP for SLP Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>On</td>
</tr>
<tr>
<td>Valid Values</td>
<td>On/Off</td>
</tr>
</tbody>
</table>

**Wait before Giving Up on DA:** Timeout (in seconds) for an SLP Request to a DA. This parameter is not used to time out requests to SAs because there is a separate setting for that.

<table>
<thead>
<tr>
<th>Table 80</th>
<th>Wait before Giving Up on DA Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>5</td>
</tr>
<tr>
<td>Valid Values</td>
<td>1 - 60,000</td>
</tr>
</tbody>
</table>

**Wait Before Registering on Passive DA:** If an SA running on the workstation receives an unsolicited DA Advertisement (that is, either the DA just started or the DA issued a heartbeat), the SA will need to register whatever services it offers. This parameter is used to specify a range that the SAs will attempt to register their services within to prevent the SAs on a network from all attempting to register with the DA at the same time. As mentioned earlier, the client workstation might need to use SLP to advertise Services it provides. This is unusual, but it might change in the future as applications begin to take advantage of this new advertising method.

<table>
<thead>
<tr>
<th>Table 81</th>
<th>Wait before Registering on Passive DA Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>2 seconds</td>
</tr>
<tr>
<td>Valid Values</td>
<td>1 to 60,000 seconds</td>
</tr>
</tbody>
</table>
The Novell Directory Agent

The Service Location Protocol (SLP) Directory Agents support SLP 1. Enhanced features let network administrators better control the collection and dissemination of network service information through SLP.

Table 82 Directory Agent Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>NetWare</th>
<th>Windows NT/2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory-enabled operation</td>
<td>Directory mode uses eDirectory to store SLP service information. This leverages existing eDirectory standards for configuring eDirectory tree structures, for a central point of administration, and for the ability of eDirectory to replicate service information. eDirectory replication services allow Directory Agent-to-Directory Agent communication. This is unique in SLP implementations and it facilitates global distribution of SLP database information. eDirectory replica services give the Directory Agent the ability to access global services from a local replica. In Directory mode, you use ConsoleOne.</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Local mode</td>
<td>Standalone operation. The SLP Directory Agent operates without using eDirectory. This lets network administrators use SLP Directory Agents in network segments that need the performance but don’t need to share the service information globally (Windows NT Directory Agent Only). Use the SLP Directory Agent property pages on the Windows NT or Windows 2000 computer. For more information, see “Managing Properties for Local Mode” on page 504.</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

494 Novell eDirectory 8.7 Administration Guide
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>NetWare</th>
<th>Windows NT/2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private mode</td>
<td>When operating in Private mode, the SLP Directory Agent accepts only SLP service registrations and requests from SLP agents configured with the SLP Directory Agent's IP address. In Private mode, the SLP Directory Agent does not multicast its presence on the network and does not answer multicast requests. For more information, see “Setting Up Private Mode” on page 506.</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Proxy scope support</td>
<td>The SLP Directory Agent acts as proxy for scopes hosted by other SLP Directory Agents. This lets network administrators distribute service information from other SLP scopes, usually not visible to a local network segment, without having to enable network directory support. For more information, see “Managing a Proxy Scope” on page 505.</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Service filtering support</td>
<td>The SLP Directory Agent can be configured with service filters that control the service information to and from SLP agents in the network. Additional filters can control the SLP service information that is stored in the network directory for global distribution. These filters provide single-point administration of the services made available through the SLP (Windows NT/2000 Directory Agent only). For more information, see “Managing Scope Filters” on page 506.</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Using the Novell Windows NT Directory Agent

Scopes

In SLP, a scope is simply a list of SLP services that have been registered with a Directory Agent.

Using Scopes in Directory Mode

In Directory mode, when a Directory Agent is created, it registers the SLP Scope Unit container object, which is the actual storage container for SLP service information. Each Scope Unit container holds all the SLP Service objects for a specific scope. You can replicate this container into other partitions within the tree or within federated trees.

As mentioned earlier, the Scope Unit has an attribute called the Scope Name. This Scope Name is used by the Service Agent and User Agent to define what scopes they are to work with. SLP scopes allow network administrators to organize SLP services into groups. The Service Agent determines which groupings the services on that the server will be registered in. By default, all SLP services are registered in the unscoped scope. When clients send SLP requests to a Directory Agent, they can specify a scope for the Directory Agent to use in order to find the service they are looking for. If no scope is specified by the client, the Directory Agent looks in the Unscoped table to find the requested service.

A Directory Agent can service multiple scopes and a Service Agent can register services in multiple scopes. The registered services can be replicated between sites by using eDirectory.

Using Scopes in Local Mode

Scopes configured in Local mode operate similarly to scopes configured for Directory mode with the exception that the scopes are stored locally instead of in eDirectory. By default, all SLP services are registered in the unscoped scope. We recommend that you configure at least one scope.

For more information about configuring scopes in Local mode, see “Managing Scopes” on page 504.

Using Scopes to Handle the 64 KB Limitation Issue

A total of 64 KB of data is the most that the Directory Agent can send to the client via a TCP connection. If there is more than 64KB of a certain service...
type, the list will be cut short. The reason for this is that in SLP 1, the length field in the SLP response packet header is only 16 bits, allowing up to 64 KB of service data.

Table 83 lists the common service types that fit into a 64 KB response packet.

<table>
<thead>
<tr>
<th>Service</th>
<th>Number per Response Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDAP.Novell</td>
<td>About 1,200, depending on the length of the partition names</td>
</tr>
<tr>
<td>Bindery.Novell</td>
<td>700 to 1,100, depending on the length of the server names</td>
</tr>
<tr>
<td>MGW.Novell</td>
<td>About 1,200</td>
</tr>
<tr>
<td>SapSrvm.Novell</td>
<td>No more than 540</td>
</tr>
</tbody>
</table>

Understanding Scope Filtering

SLP uses scopes to logically group services according to administration, usage, or service type criteria. By dictating the scopes that SLP User Agents and Service Agents participate in, you can control the service information users see. Unfortunately, that level of control is not sufficient for large and sophisticated network environments. To give you better control over the collection and distribution of service information, use the additional filtering capabilities provided as part of the SLP Directory Agent configuration and management tools.

When administering scopes, you can configure registration, response, and directory filters for each scope.

- Registration filters restrict or control the service information that is accepted and stored by the Directory Agent for a given scope.
- Response filters restrict or control the service information that is returned to specific users or groups of users.
- Directory filters control whether the service information that is registered with the Directory Agent (subject to the registration filters) is also stored in the corresponding Scope Unit container object.

The Registration, Response, and Directory filters are configured on a per-scope basis. This lets you separately control the type of information stored in each scope.
Filtering

The SLP Directory Agent can be configured with service filters that control the service information to and from SLP agents in the network. Additional filters can control the SLP service information that is stored in the network directory for global distribution. These filters provide single-point administration of the services made available through SLP (Windows NT/2000 Directory Agent only).

Using INCLUDE and EXCLUDE Filters

Registration, response, and directory filters are specified using the INCLUDE and EXCLUDE filter directives.

The INCLUDE filter directive specifies criteria that the service registration or request must comply with to store or retrieve service information in the specified scope.

The EXCLUDE filter directive specifies criteria that prohibit any compliant service registration or request from occurring for the specified scope.

The filters associated with a scope consist of one or more INCLUDE and EXCLUDE filter directives. For a service registration or request to be processed, it must match at least one INCLUDE filter directive and not match any EXCLUDE filter directives configured for the scope. If any INCLUDE directives are configured, only service registrations and requests matching at least one INCLUDE directive are processed; all others are denied. If no INCLUDE directives are configured, all service registrations and requests are processed subject to any EXCLUDE filter directives.

The criteria for an INCLUDE or EXCLUDE filter directive are specified by one or more filter operations. Filter operations allow the administrator to filter the type of service, the specific service URLs, a service registration's lifetime, or the address of the sending or requesting host in the network. If you specify multiple filter operations in a single filter directive, all filter operations must evaluate to TRUE for the filter directive to be TRUE. Only one of each type of filter operation can be included in a single filter directive.

If the IP address of the sending or requesting host is used as filter criteria, it is specified in dotted decimal notation (for example, 137.65.143.195). Subnet masks can be associated with an IP address by appending a slash (/) followed by the subnet mask. The subnet mask can be specified using either dotted decimal notation or by specifying the number of contiguous bits that constitute the mask (for example, 137.65.143.0/255.255.252.0 and 137.65.143.0/22 are equivalent). If a subnet mask is specified, the mask will be applied to both the...
address specified in the ADDRESS filter operation and the host IP address will be checked before any filter evaluations are made.

**Filter Syntax**

The ABNF (RFC 2234) for the registration, response, and directory filters is defined below:

Registration Filter = 1*(include_directive / exclude_directive)  
Response Filter = 1*(include_directive / exclude_directive)  
Directory Filter = 1*(include_directive / exclude_directive)  
include_directive = "INCLUDE("filter_operation")"  
exclude_directive = "EXCLUDE("filter_operation")"  
filter_operation = [address_operation] [type_operation] [lifetime_operation] [url_operation]  
address_operation = "/(ADDRESS equality_operator *1(ipv4_number / ipv4_number "/" subnet_mask ")")"  
lifetime_operation = "/(LIFETIME filter_operator seconds")"  
type_operation = "/(TYPE equality_operator [wild] service_type [wild])"  
url_operation = "/(URL equality_operator [wild] service_url [wild])""  
service_url = service: URL as defined by RFC 2609  
service_type = abstract-type ":" url_scheme / concrete-type  
abstract_type = type_name ["." naming_auth]  
concrete_type = protocol ["." naming_auth]  
type_name = resname  
naming_auth = resname  
protocol = resname  
url-scheme = resname  
wild = "*"  
reserved = "( / )" / "*" / "\"  
escaped = "\" reserved  
resname = ALPHA [1*(ALPHA / DIGIT / "+" / "-" )]  
ipv4_number = 1*3DIGIT 3("." 1*3DIGIT)  
subnet_mask = ipv4_number / 1-32  
equality_operator = ":=" / ":="  
filter_operator = ":=" / ":=" / ":=" / ":<"  
seconds = 1-65535
Examples of INCLUDE and EXCLUDE Filter Directives

Following are examples of INCLUDE and EXCLUDE filter directives to help you understand how to implement the filter feature.

- Registration Filters

  Allow only services of types ndap.novell or bindery.novell with a lifetime greater than 5,000 seconds from servers on the 137.65.140.0 subnet to be stored by the SLP Directory Agent. The ADDRESS operation values for both INCLUDE directives are equivalent. The first registration filter uses dotted decimal notation for the subnet address and the second registration filter specifies the number of bits in the subnet mask.

  INCLUDE((TYPE == ndap.novell)(ADDRESS == 137.65.140.0/255.255.252.0))
  INCLUDE((TYPE == bindery.novell)(ADDRESS == 137.65.140.0/22))
  EXCLUDE ((LIFETIME < 5000))

- Response Filters

  Prevent only workstations on the 137.65.140.0 subnet (except the workstation with IP address 137.65.143.155) from accessing information held by the SLP Directory Agent.

  INCLUDE((ADDRESS == 137.65.140.0/255.255.252.0))
  EXCLUDE((ADDRESS == 137.65.143.155))

- Directory Filters

  The first two directory filters allow only services of types ndap.novell and bindery.novell to be stored in the Scope Unit container object associated with this scope. The second two directory filters allow only services with the URLs specified to be stored in the Scope Unit container object associated with this scope.

  INCLUDE((TYPE == ndap.novell))
  INCLUDE (TYPE == bindery.novell)
  or
  INCLUDE((URL == service:ndap.novell://GLOBAL_PARTITION1.CORP_TREE,))
  INCLUDE (URL == service:ndap.novell://GLOBAL_PARTITION2.CORP_TREE)
When the Directory Agent is operating in Local mode, the registration, response, and directory filters are stored in the local system’s registry and are persistent across system boots.

When the Directory Agent is operating in Directory mode, the registration, response, and directory filters are stored as part of the Scope Unit directory object defining the filtered scope. The Scope Unit object has a Registration Filters, Response Filters, and Directory Filters attribute. These attributes are multivalued of type SYN_CI_STRING. Each INCLUDE and EXCLUDE filter directive is stored as a separate string in the Registration Filters, Response Filters, or Directory Filters attribute.

Using the Service Location Protocol Directory Agent

The following scenarios show some of the many options for deploying SLP.

- “Scenario 1: Remote Site with a Mixed NetWare and Windows NT Environment” on page 501
- “Scenario 2: Remote Office with Windows NT Servers Only” on page 502
- “Scenario 3: Using the Directory Agent for a Small Group of Users” on page 502
- “Scenario 4: Restricting SLP Information” on page 502
- “Scenario 5: Synchronizing SLP Information over a WAN Link” on page 502
- “Scenario 6: Replicating SLP Information to a Remote Site” on page 502
- “Scenario 7: Running a Directory Agent in Local Mode” on page 503
- “Scenario 8: Using the Proxy Feature” on page 503

Scenario 1: Remote Site with a Mixed NetWare and Windows NT Environment

**Situation:** A remote office is running NT servers and NetWare clients with no NetWare servers. The administrator wants the clients to see all the network services from a local server, avoiding sending on-demand service queries over the slow link.

**Solution:** The Directory Agent can be installed on a Windows NT server to allow the clients to see all the network services from a local server without causing on-demand traffic over the slow link.
Scenario 2: Remote Office with Windows NT Servers Only

**Situation:** A remote office is running NT servers, and the administrator wants local clients to see only a limited set of services.

**Solution:** Use the new Directory Agent and its filter or proxy capabilities to configure the Directory Agent to see only a specific set of services.

Scenario 3: Using the Directory Agent for a Small Group of Users

**Situation:** An administrator wants to configure a Directory Agent for a small group of users and wants that Directory Agent to manage only a small subset of services, not all SLP services on the network.

**Solution:** The administrator defines exactly which services are allowed to register with that Directory Agent. Then, by statically assigning the Directory Agent's address to those users, the administrator controls which services are seen by those users.

Scenario 4: Restricting SLP Information

**Situation:** An administrator wants to restrict the users who can query SLP information from a Directory Agent.

**Solution:** Set the filters on the Directory Agent for Windows NT to define who can obtain information from the Directory Agent. This identification is determined by the IP address.

Scenario 5: Synchronizing SLP Information over a WAN Link

**Situation:** An administrator wants to synchronize SLP service information over a WAN link, but one side of the link uses eDirectory without any NetWare servers.

**Solution:** Run the Directory Agent on a Windows NT server and configure the Directory Agent to service the eDirectory scope containers included in the network eDirectory replication design.

Scenario 6: Replicating SLP Information to a Remote Site

**Situation:** An administrator wants to replicate SLP service data to a remote site without using eDirectory as the replication method.

**Solution:** The Directory Agent is installed on a Windows NT server at the remote site and is configured to proxy the data in another Directory Agent's...
scope. The Directory Agent scope that contains the original service information is known as the Scope Authority. The Directory Agent at the remote site is configured to look at a Scope Authority and can replicate the data to the remote site by using standard SLP requests to the Directory Agent.

Scenario 7: Running a Directory Agent in Local Mode

**Situation:** An administrator needs SLP on the network to find printers and other services. He also needs a Directory Agent to handle unicast requests because multicast packets are disabled on the network and unicast is more efficient in bandwidth use.

**Solution:** Run the Directory Agent for Windows NT in a Local mode of operation. The services are only stored in memory and not in a Directory Service. This means that the Directory Agent can be run on Windows NT without the Novell Client or eDirectory.

Scenario 8: Using the Proxy Feature

**Situation:** An administrator of a development group notices that services are going up and down. He wants a more active method of making sure the service information in SLP is accurate instead of relying on the default service lifetime protocol.

**Solution:** Use the proxy feature in the Directory Agent for Windows NT to configure the Directory Agent to poll another Directory Agent or Service Agents’s scope. Configure the Directory Agent with Service Agent IP addresses as the Scope Authorities. This causes the Directory Agent to poll each Service Agent at a configured interval, querying for all active services.

Setting Up SLP on Windows NT or Windows 2000

This section explains how to set up SLP on a Windows NT or Windows 2000 system.

- “Installing the Windows NT/Windows 2000 Directory Agent” on page 504
- “Managing Properties for Local Mode” on page 504
- “Managing Scopes” on page 504
- “Managing a Proxy Scope” on page 505
- “Managing Scope Filters” on page 506
Installing the Windows NT/Windows 2000 Directory Agent

1. On a Windows NT or Windows 2000 machine, insert the Novell eDirectory CD.
2. Click Start > Run > Browse.
3. Select Setup.exe from the \NT directory on the CD.
4. From the installation screen, click SLP Directory Agent > Install.
   Follow the online instructions for the SLP Directory Agent installation program.
   If you select a Directory type of setup, the eDirectory schema will be extended for the eDirectory tree specified.

Managing Properties for Local Mode

1. On the server where the Directory Agent is running, click Start > Programs > Novell SLP Directory Agent > SLP DA Properties.
2. Adjust the configuration properties for the Directory Agent.

Managing Scopes

To add, delete, or modify a scope:

1. On the server where the Directory Agent is running, click Start > Programs > Novell SLP Directory Agent > SLP DA Properties.
2. Click Scopes > Add. Then enter the name of the new Scope, then click OK,
   or
   Select an existing scope, then click Properties to modify the scope or Delete to delete the scope from the list.

For more information about setting up scope filtering, see “Managing a Proxy Scope” on page 505.

For more information about setting up a proxy scope, see “Managing Scope Filters” on page 506.
Managing a Proxy Scope

To add or delete a scope:

1. On the server where the Directory Agent is running, click Start > Programs > Novell SLP Directory Agent > SLP DA Properties.
2. Click Scopes, then select the scope you want to add a proxy to from the list.
3. Click Properties > Proxy.
4. Enter the name of the Scope Authority to be proxied, then click Add.

The syntax of the Scope Authority specification is as follows:

```
scope_authority [/refresh_interval] [/character_encoding] [/target_scope]
```

The variables for this syntax include the following:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Scope_authority</code></td>
<td>The IP address or DNS name of the Directory Agent acting as the scope authority for the proxied scope.</td>
</tr>
<tr>
<td><code>Refresh_interval</code></td>
<td>The amount of time (in minutes) service information from this scope authority is to be retrieved. This value overrides any other refresh interval time configured for the scope but only applies to this scope authority.</td>
</tr>
<tr>
<td><code>Character_encoding</code></td>
<td>Indicates the character encoding to use when retrieving service information from this scope authority. Character encodings include ASCII, UTF8, and Unicode.</td>
</tr>
<tr>
<td><code>Target_scope</code></td>
<td>The name of the scope to be queried for service information. If this value is omitted, the current scope name is used.</td>
</tr>
</tbody>
</table>
Managing Scope Filters

To add, delete, or modify scope filters:

1. On the server where the Directory Agent is running, click Start > Programs > Novell SLP Directory Agent > SLP DA Properties.
2. Click Scopes, then select the scope you want to add the filter to from the list.
3. Click Properties > Filters.
4. Select the type of filter you want to add, then click Add.
5. Select the filter parameters you want to include and exclude.

For more information about the filtering, see “Understanding Scope Filtering” on page 497.

Setting Up Private Mode

Private mode allows you to limit the visibility of the Directory Agent to those services configured specifically with the Directory Agent’s IP address.

1. On the server where the Directory Agent is running, click Start > Programs > Novell SLP Directory Agent > SLP DA Properties.
2. Check or uncheck the Private Mode check box to enable or disable Private mode.

Managing the Directory Agent in Directory Mode with ConsoleOne

To manage a directory agent running in Directory mode, use ConsoleOne. You can install ConsoleOne from the eDirectory CD.

- “Setting Configuration Properties in Directory Mode” on page 506
- “Adding a Serviced Scope Unit” on page 507
- “Setting Up Scope Filters” on page 507

Setting Configuration Properties in Directory Mode

1. In ConsoleOne, double-click the SLP Directory Agent object you want to modify.

   The Properties page for the Directory Agent appears.

2. Adjust the settings.
Adding a Serviced Scope Unit

1. In ConsoleOne, double-click the SLP Directory Agent object you want to modify.
   The Properties page for the Directory Agent appears.
2. Click the SLP Scope Units tab, then click Add.
3. Browse the eDirectory tree, then select the Scope Unit you want to add.

Setting Up Scope Filters

1. In ConsoleOne, right-click the SLP Scope Unit object, then click Properties.
   The Properties page for the SLP Scope Unit appears.
2. Click Filters, then select the type of filter you want to create.
3. Click Add, then enter the filter information you want.
   For more information about filtering, see “Managing Scope Filters” on page 506.

Setting Up SLP on NetWare

This section explains how to set up SLP on a NetWare Server.

- “Installing the NetWare SLP Directory Agent” on page 507
- “Setting Up the NetWare Directory Agent Manually” on page 508
- “NetWare SLP Directory Agent Console Commands” on page 508
- “NetWare SLP Directory Agent SET Commands” on page 510

Installing the NetWare SLP Directory Agent

The software to implement SLP on NetWare is installed on the server during the server installation.

1. At the server console, enter LOAD SLPDA.
   The program searches eDirectory for a SLP Directory Agent. If an agent does not exist, the program reports that a SLP Directory Agent has not been configured.
2. Press Enter to set up a default configuration.
The schema is extended and a Directory Agent object with the name `server_name_SLPDA` and a Scope Unit named `SLP_SCOPE` are created and linked. This is recommended unless you want to create the SLP objects in eDirectory yourself.

Use ConsoleOne to adjust settings for the Directory Agent object.

### Setting Up the NetWare Directory Agent Manually

To configure SLP using ConsoleOne:

1. Select the container where you want the SLPDA to reside.
2. Click Object > Create > SLP Directory Agent > OK.
3. Enter the Directory Agent object name, click Define Additional Properties, then click Create.
4. Select a host server, then select the container where you want the Scope Units stored.
5. Click Object > Create > SLP Scope Unit > OK.
6. Type the name for the SLP Scope Unit.
7. Double-click the SLP Directory Agent object.
8. Click the SLP Scope Units page, then click Add.
9. Select the scope units serviced by this Directory Agent.

### NetWare SLP Directory Agent Console Commands

<table>
<thead>
<tr>
<th>SLP Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLP OPEN filename.log</td>
<td>Creates the SLP trace file in the root of volume sys:</td>
</tr>
<tr>
<td>SLP CLOSE</td>
<td>Closes the SLP trace file.</td>
</tr>
</tbody>
</table>
Common Novell SLP service types include the following:

- BINDERY.NOVELL (NetWare servers)
- CMD.NOVELL (Compatibility mode server/relay agents)
- DIRECTORY-AGENT (sends an SLP multicast packet to rediscover the DA on the network)
- MGW.NOVELL (Compatibility mode gateway/migration agents)
- NDAP.NOVELL (NDS)
- RCONSOLE.NOVELL (Java* RCONSOLE)
- RMS.NOVELL (Resource Management Service of NDPS®)
- SRS.NOVELL (NDPS broker)
- SAPSRV.NOVELL (NetWare 5 or later servers with IPX CMD loaded)

SLP restrictions are as follows:

\[ slp\_attribute==value \]

Other operators available are <=, and >=.

Examples of using the Display SLP Services command include the following:

- DISPLAY SLP SERVICES (Displays all known SLP services)
- DISPLAY SLP SERVICES BINDERY.NOVELL (Displays all bindery.novell services)
- DISPLAY SLP SERVICES MBW.NOVELL/(CMD NETWORK==ABC12345)/ (Displays all the Migration Agents servicing the CMD network number ABC12345)
- DISPLAY SLP SERVICES BINDERY.NOVELL// (SVCNAME-WS==ABC*)/ (Displays bindery.novell services with names that begin with abc)
- DISPLAY SLP SERVICES BINDERY.NOVELL/PROVO/ (SVCNAME-WS==ABC*)/ (Displays bindery.novell services with names that begin with abc in scope provo)
The following is an example of using the Display SLP attributes command:

```
DISPLAY SLP ATTRIBUTES SERVICE:BINDERY.NOVELL:///SERVER1 (Displays all SLP attributes and values for the SERVER1 bindery.novell object)
```

**NetWare SLP Directory Agent SET Commands**

**Table 86  SLP Directory Agent SET Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| SET SLP DA Discovery Options = value | 0x01 = Use multicast DA advertisements  
0x02 = Use DHCP discovery  
0x03  
0x04 = Use static file sys:etc\slp.cfg  
0x05  
0x06  
0x07  
0x08 = Scopes Required  
Value = 0 to 8  
Default = 3 |
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET SLP TCP = value</td>
<td>This sets SLP to use TCP packets instead of UDP packets when possible.</td>
</tr>
<tr>
<td></td>
<td>Value = ON/OFF</td>
</tr>
<tr>
<td></td>
<td>Default = OFF</td>
</tr>
<tr>
<td>SET SLP DEBUG = value</td>
<td>0x01 = COMM</td>
</tr>
<tr>
<td></td>
<td>0x02 = TRAN</td>
</tr>
<tr>
<td></td>
<td>0x04 = API</td>
</tr>
<tr>
<td></td>
<td>0x08 = DA</td>
</tr>
<tr>
<td></td>
<td>0x010 = ERR</td>
</tr>
<tr>
<td></td>
<td>0x020 = SA</td>
</tr>
<tr>
<td></td>
<td>These bits can be combined with AND or OR statements for multiple values. An example of COMM and API would be 0x05.</td>
</tr>
<tr>
<td></td>
<td>Value = 0 to 4294967255</td>
</tr>
<tr>
<td></td>
<td>Default = 0</td>
</tr>
<tr>
<td>SET SLP Multicast Radius = value</td>
<td>Specifies an integer describing the multicast radius.</td>
</tr>
<tr>
<td></td>
<td>Value = 0 to 32</td>
</tr>
<tr>
<td></td>
<td>Default = 32</td>
</tr>
<tr>
<td>SET SLP Broadcast = value</td>
<td>Sets the use of broadcast packets instead of multicast packets.</td>
</tr>
<tr>
<td></td>
<td>Value = ON/OFF</td>
</tr>
<tr>
<td></td>
<td>Default = OFF</td>
</tr>
<tr>
<td>SET SLP MTU size= value</td>
<td>Specifies an integer describing the maximum transfer unit size.</td>
</tr>
<tr>
<td></td>
<td>Value = 0 to 4294967255</td>
</tr>
<tr>
<td></td>
<td>Default = 1472</td>
</tr>
</tbody>
</table>

Implementing the Service Location Protocol
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| SET SLP Rediscover Inactive Directory Agents = value | Specifies the minimum time period in seconds that SLP will wait to issue service requests to rediscover inactive directory agents.  
  Value = 0 to 4294967255  
  Default = 60 |
| SET SLP Retry Count = value | Specifies an integer value describing the maximum number of retries.  
  Value = 0 to 128  
  Default = 3 |
| SET SLP Scope List = value | Specifies a comma-delimited scope policy list.  
  Value = 1023 maximum  
  Default = 1023 |
| SET SLP SA Default Lifetime = value | Specifies an integer value describing the default lifetime in seconds of service registers.  
  Value = 0 to 4294967255  
  Default = 900 |
| SET SLP Event Timeout = value | Specifies an integer value describing how long (in seconds) to wait before timing out multicast packet requests.  
  Value = 0 to 4294967255  
  Default = 53 |
| SET SLP DA Heart Beat Time = value | Specifies an integer value describing how long (in seconds) to wait before sending the next Directory Agent heartbeat packet.  
  Value = 0 to 4294967255  
  Default = 10800 |
| SET SLP Close Idle TCP Connections Time = value | Specifies an integer value describing how long (in seconds) to wait before terminating idle TCP connections.  
  Value = 0 to 4294967255  
  Default = 300 |
| SET SLP DA Event Timeout = value | Specifies an integer value describing how long (in seconds) to wait before timing out Directory Agent packet requests.  
  Value = 0 to 429  
  Default = 5 |
Setting Up SLP on Linux or Solaris

This section contains the following information:

- “User Agents and Service Agents” on page 514
- “Starting and Stopping the Daemon Process” on page 515
- “Using the SLPINFO Diagnostic Utility” on page 516
- “eDirectory Interoperability with OpenSLP on Linux and Solaris 8.0 SLP” on page 516
- “SLP V1- V2 Interoperability Issues” on page 517

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET SLP Maximum WTD = value</td>
<td>Specifies the maximum number of work-to-do threads that SLP can allocate.</td>
</tr>
<tr>
<td></td>
<td>Value = 1 to 64</td>
</tr>
<tr>
<td></td>
<td>Default = 10</td>
</tr>
<tr>
<td>SET SLP Reset = value</td>
<td>Forces the SA to send new service registers and forces the SA to send DA Advertise packets. (Resets to OFF each time it is set to ON)</td>
</tr>
<tr>
<td></td>
<td>Value = ON</td>
</tr>
<tr>
<td>SET SLP Debug = value</td>
<td>0x01 = COMM</td>
</tr>
<tr>
<td></td>
<td>0x02 = TRAN</td>
</tr>
<tr>
<td></td>
<td>0x04 = API</td>
</tr>
<tr>
<td></td>
<td>0x08 = SA_DA</td>
</tr>
<tr>
<td></td>
<td>0x08 = ERR</td>
</tr>
<tr>
<td></td>
<td>0x020 = SA</td>
</tr>
<tr>
<td></td>
<td>0x040 = UA_DA</td>
</tr>
<tr>
<td></td>
<td>These bits can be combined with AND or OR statements for multiple values. An example of COMM and API would be 0x05.</td>
</tr>
<tr>
<td></td>
<td>Value = 0 to 65535</td>
</tr>
<tr>
<td></td>
<td>Default = 88</td>
</tr>
</tbody>
</table>
User Agents and Service Agents

The slpuasa daemon provides the functionality for both the User Agent and the Service Agent the daemon will read the /etc/slpuasa.conf file on start-up and perform an Active-Discovery of the DA on the network if the isMulticastOnly option is disabled or commented out (the default value is 0). Otherwise, the daemon will use the multicast option.

The slpuasa.conf file contains a list of configuration parameters for configuring the SLP User Agent - Service Agent daemon. The slpuasa daemon reads this file when it is started. If any change is made to the slpuasa.conf file, the daemon needs to be restarted for the changes to be reflected.

Each parameter in the slpuasa.conf file is in a single line in the form, as follows:

```
parameter=value
```

The parameters are described in Table 87.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA_ADDR</td>
<td>A list of IP addresses and scopes of SLP DAs to statically configure the User Agent/Service Agent, in the format uns scoped_da_ip_addr1, uns scoped_da_ip_addr2, ... uns scoped_da_ip_addrm, (scoped_da_ip_addr1, list of da scopes), (scoped_da_ip_addr2, list of da scopes), ... (scoped_da_ip_addrm, list of da scopes)</td>
</tr>
<tr>
<td>net.slp.isBroadcastOnly</td>
<td>Lets you to select either broadcast or multicast SLP requests. A value of 0 means the User Agent and Service Agent use multicast. A value of 1 means they use broadcast. Default = 0.</td>
</tr>
</tbody>
</table>

514 Novell eDirectory 8.7 Administration Guide
Starting and Stopping the Daemon Process

The slpuasa can be started and stopped with the slpuasa script.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris</td>
<td>/etc/init.d/slpuasa {start/stop}</td>
</tr>
<tr>
<td>Linux</td>
<td>/etc/rc.d/init.d/slpuasa {start/stop}</td>
</tr>
</tbody>
</table>

Parameter | Description
---|---
net.slp.isMulticastOnly | Use this option only if you have problems because of a Directory Agent, because this parameter can cause an unnecessary increase in the multicast traffic in the network.
A value of 1 means that the slpuasa daemon will not discover any Directory Agents (active or passive).
Default = 0

net.slp.MTU | The Maximum Path Unit to be set parameter is used by the Service Agent to service the SLP requests. The default.
Default = 1400

net.slp.MulticastRadius | The site’s multicast TTL.
Default = 32

net.slp.useScopes | List of strings indicating the scopes the User Agent/Service Agent is allowed to use when making requests or registering.
Using the SLPINFO Diagnostic Utility

The slpinfo diagnostic utility is used to list the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a &quot;service:ndap.novell:///treename.&quot;</td>
<td>Lists the attributes of the respective tree, such as svcname, svcaddr (the TCP and UDP addresses of the Server), svcid, version, and nds-version.</td>
</tr>
<tr>
<td>-s variable</td>
<td>Lists various services under the service-type. For example, slpinfo -s &quot;ndap.novell:///&quot; will list all the services under the Service-Type &quot;ndap.novell.&quot; The query can also be used with attributes. For example, slpinfo -s &quot;ndap.novell//svcname-ws==<em>name</em>.)&quot; will list all the services containing the text &quot;name&quot;.</td>
</tr>
<tr>
<td>-t</td>
<td>Lists the various service-types.</td>
</tr>
</tbody>
</table>

eDirectory Interoperability with OpenSLP on Linux and Solaris 8.0 SLP

Novell SLP (V1) is now an optional package.

If Caldera OpenSLP RPMs and Solaris 8.0 SLP packages are already installed on Linux and Solaris respectively, the eDirectory installation will skip SLP install.

eDirectory uses the platform specific SLP API’s by default.

To use Novell SLP(v1) on a system that has another SLP package from a different vendor, go to the setup directory of eDirectory and do the following:

1. To install Novell SLP enter the following command:
   - rpm -ivh NDSslpxxx (For Linux)
   - pkgadd -d NDSslpxxx (For Solaris)

2. Ensure that you select the appropriate SLP daemon to use in the startup scripts.
SLP V1- V2 Interoperability Issues

A network should have SLPv2 DA for compatibility issues between SLPv1 and SLPv2 hosts, because SLPv1 UAs will not receive replies from SLPv2 SAs and SLPv2 UAs will not receive replies from SLPv1 SAs.