Managing Service Location Protocol Services in Oracle® Solaris 11.1

Part No: E28998–01
October 2012

ORACLE

121010@25097

Contents

Preface 13

1 SLP (Overview) 15
SLP Architecture 15
Summary of the SLP Design 16
SLP Agents and Processes 16
SLP Implementation 18
Other SLP Information Sources 19

2 Planning and Enabling SLP (Tasks) 21
SLP Configuration Considerations 21
Deciding What to Reconfigure 22
Using `snoop` to Monitor SLP Activity 22
▼ How to Use `snoop` to Run SLP Traces 23
Analyzing a `snoop slp` Trace 23

3 Administering SLP (Tasks) 27
Configuring SLP Properties 27
SLP Configuration File: Basic Elements 28
▼ How to Change Your SLP Configuration 29
Modifying DA Advertising and Discovery Frequency 30
Limiting UAs and SAs to Statically Configured DAs 30
▼ How to Limit UAs and SAs to Statically Configured DAs 30
Configuring DA Discovery for Dial-up Networks 31
▼ How to Configure DA Discovery for Dial-up Networks 31
Configuring the DA Heartbeat for Frequent Partitions 33
▼ How to Configure DA Heartbeat for Frequent Partitions 33

Relieving Network Congestion 34
Accommodating Different Network Media, Topologies, or Configurations 34
Reducing SA Reregistrations 34
▼ How to Reduce SA Reregistrations 35
Configuring the Multicast Time-to-Live Property 35
▼ How to Configure the Multicast Time-to-Live Property 36
Configuring the Packet Size 37
▼ How to Configure the Packet Size 37
Configuring Broadcast-Only Routing 38
▼ How to Configure Broadcast-Only Routing 38
Modifying Timeouts on SLP Discovery Requests 39
Changing Default Timeouts 39
▼ How to Change Default Timeouts 40
Configuring the Random-Wait Bound 41
▼ How to Configure the Random-Wait Bound 41
Deploying Scopes 42
When to Configure Scopes 43
Considerations When Configuring Scopes 44
▼ How to Configure Scopes 44
Deploying DAs 45
Why Deploy an SLP DA? 45
When to Deploy DAs 47
▼ How to Deploy DAs 47
Where to Place DAs 48
SLP and Multihoming 49
Multihoming Configuration for SLP 49
When to Configure for Nonrouted, Multiple Network Interfaces 49
Configuring Nonrouted, Multiple Network Interfaces (Task Map) 50
Configuring the `net.slp.interfaces` Property 50
Proxy Advertising on Multihomed Hosts 52
DA Placement and Scope Name Assignment 52
Considerations When Configuring for Nonrouted, Multiple Network Interfaces 53

4 Incorporating Legacy Services 55
When to Advertise Legacy Services 55

Advertising Legacy Services 55
Modifying the Service 55
Advertising a Service That Is Not SLP Enabled 56
SLP Proxy Registration 56
▼ How to Enable SLP Proxy Registration 56
Using SLP Proxy Registration to Advertise 57
Considerations When Advertising Legacy Services 59

5 SLP (Reference) 61
SLP Status Codes 61
SLP Message Types 62

Index 65

Figures

FIGURE 1–1 SLP Basic Agents and Processes 17
FIGURE 1–2 SLP Architectural Agents and Processes Implemented With a DA 17
FIGURE 1–3 SLP Implementation 19

Tables

TABLE 1–1 SLP Agents 16
TABLE 3–1 SLP Configuration Operations 27
TABLE 3–2 DA Advertisement Timing and Discovery Request Properties 30
TABLE 3–3 SLP Performance Properties 34
TABLE 3–4 Time-out Properties 39
TABLE 3–5 Configuring Nonrouted, Multiple Network Interfaces 50
TABLE 4–1 SLP Proxy Registration File Description 58
TABLE 5–1 SLP Status Codes 61
TABLE 5–2 SLP Message Types 62

Examples

EXAMPLE 3–1 Setting up `slpd` to Operate as a DA Server 29

Preface

Managing Service Location Protocol Services in Oracle Solaris 11.1 is part of a multivolume set that covers a significant part of the Oracle Solaris system administration information. This book assumes that you have already installed the Oracle Solaris operating system, and you have set up any networking software that you plan to use.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that run the Oracle Solaris release. To use this book, you should have one to two years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Description	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

◆ ◆ ◆ CHAPTER 1

SLP (Overview)

The Service Location Protocol (SLP) provides a portable, platform-independent framework for the discovery and provisioning of SLP-enabled network services. This chapter describes the SLP architecture and the Oracle Solaris implementation of SLP for IP intranets.

- "SLP Architecture" on page 15
- "SLP Implementation" on page 18

SLP Architecture

This section outlines the fundamental operation of SLP and describes agents and processes that are used in SLP administration.

SLP provides all of the following services automatically, with little or no configuration.

- Client application requests for information that is required to access a service
- Advertisement of services on network hardware devices or software servers; for example, printers, file servers, video cameras, and HTTP servers
- Managed recovery from primary server failures

In addition, you can do the following to administer and tune SLP operation if necessary.

- Organize services and users into *scopes* that are composed of logical or functional groups
- Enable SLP logging to monitor and troubleshoot the SLP operation on your network
- Adjust SLP timing parameters to enhance performance and scalability
- Configure SLP not to send and not to process multicast messages when SLP is deployed on networks that lack support for multicast routing
- Deploy SLP Directory Agents to improve scalability and performance

Summary of the SLP Design

SLP libraries inform network-aware agents that advertise services in order for those services to be discovered over a network. SLP agents maintain up-to-date information on the type and location of services. These agents can also use proxy registrations to advertise services that are not directly SLP enabled. For more information, see Chapter 4, "Incorporating Legacy Services."

Client applications rely on SLP libraries that make requests directly to the agents that advertise services.

SLP Agents and Processes

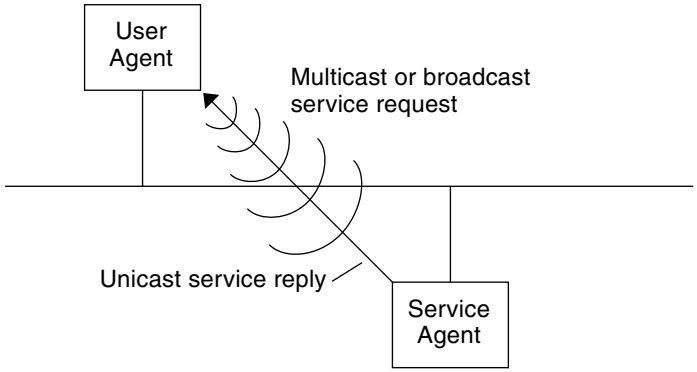
The following table describes the SLP agents.

TABLE 1–1 SLP Agents

SLP Agent	Description
Directory Agent (DA)	Process that caches SLP advertisements that are registered by Service Agents (SAs). The DA forwards service advertisements to User Agents (UAs) on demand.
Service Agent (SA)	SLP agent that acts on behalf of a service to distribute service advertisements and to register the service with Directory Agents (DAs).
User Agent (UA)	SLP agent that acts on behalf of a user or application to obtain service advertisement information.
scope	An administrative or logical grouping of services.

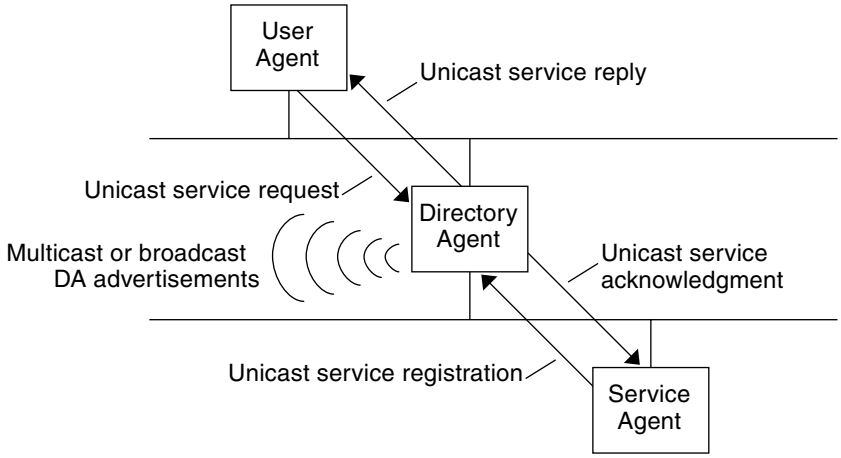
The following figure shows the basic agents and processes that implement the SLP architecture. The figure represents a default deployment of SLP. No special configuration has been done. Only two agents are required: the UA and SA. The SLP framework allows the UA to multicast requests for services to the SA. The SA unicasts a reply to the UA. For example, when the UA sends a service request message, the SA responds with a service reply message. The service reply contains the location of services that match the client's requirements. Other requests and replies are possible for attributes and service types. For more information, see Chapter 5, "SLP (Reference)."

FIGURE 1–1 SLP Basic Agents and Processes



The following figure shows the basic agents and processes that implement the SLP architecture when a DA is deployed in the framework.

FIGURE 1–2 SLP Architectural Agents and Processes Implemented With a DA



When you deploy DAs, fewer messages are sent in the network and UAs can retrieve information much faster. DAs are essential when the size of a network increases or for situations in which there is no support for multicast routing. The DA serves as a cache for registered service advertisements. SAs send register messages (`SrvReg`) that list all the services they advertise to DAs. SAs then receive acknowledgments (`SrvAck`) in reply. The service advertisements are refreshed with the DA, or they expire according to the lifetime that is set for the advertisement. After a UA discovers a DA, the UA unicasts a request to the DA rather than multicasting requests to SAs.

For more information about Oracle Solaris SLP messages, refer to Chapter 5, "SLP (Reference)."

SLP Implementation

In the Oracle Solaris SLP implementation, the SLP SAs, UAs, DAs, SA servers, scopes, and other architectural components in Table 1–1 are partially mapped into `slpd` and partially into application processes. The SLP daemon, `slpd`, organizes certain off-host SLP interactions to do the following:

- Employ passive and active directory agent discovery in order to discover all DAs on the network
- Maintain an updated table of DAs for the use of the UAs and SAs on the local host
- Act as a proxy SA server for legacy service advertisements (proxy registration)

You can set the `net.slpisDA` property to also configure `slpd` to act as a DA. See Chapter 3, "Administering SLP (Tasks)."

For more information about the SLP daemon, see `slpd(1M)`.

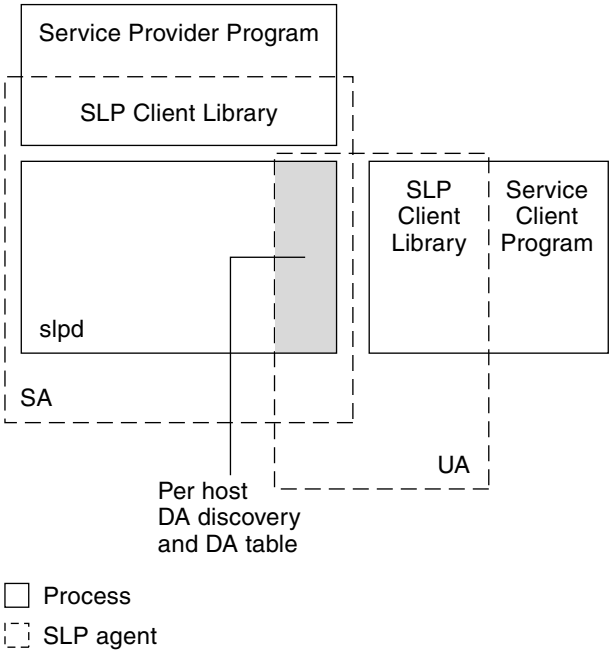
In addition to `slpd`, the C/C++ and Java client libraries (`libslp.so` and `slp.jar`) enable access to the SLP framework for UA and SA clients. The client libraries provide the following features:

- Software that offers network services which can register and deregister service advertisements
- Client software that can request services by issuing queries for service advertisements
- The list of SLP scopes available for registration and requests

No special configuration is necessary to enable the inter-process communication between `slpd` and the client libraries that provide the previous services. You must, however, run the `slpd` process first before you load the client libraries in order for the libraries to function.

In the following figure, the SLP client library in the Service Provider Program employs SA functionality. The Service Provider Program uses the SLP client library to register and deregister services with `slpd`. The SLP client library in the Service Client Program employs UA functionality. The Service Client Program uses the SLP client library to make requests. The SLP client library either multicasts requests to SAs or unicasts them to DAs. This communication is transparent to the application except that the unicast method of issuing requests is faster. The behavior of the client library can be affected by setting various SLP configuration properties. For further information, see Chapter 3, "Administering SLP (Tasks)." The `slpd` process handles all SA functionality, such as answering multicast requests and registering with DAs.

FIGURE 1–3 SLP Implementation



Other SLP Information Sources

Refer to the following documents for further information on SLP:

- Kempf, James, and Pete St. Pierre. *Service Location Protocol for Enterprise Networks*. John Wiley & Sons, Inc. ISBN Number: 0–471–31587–7.
- *Authentication Management Infrastructure Administration Guide*. Part Number: 805–1139–03.
- Guttman, Erik, Charles Perkins, John Veizades, and Michael Day. *Service Location Protocol, Version 2, RFC 2608* from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2608.txt]
- Kempf, James, and Erik Guttman. *An API for Service Location, RFC 2614* from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2614.txt]

◆ ◆ ◆ CHAPTER 2

Planning and Enabling SLP (Tasks)

This chapter provides information on planning and enabling SLP. The following sections discuss SLP configuration and the process for enabling SLP.

- "SLP Configuration Considerations" on page 21
- "Using snoop to Monitor SLP Activity" on page 22

SLP Configuration Considerations

The SLP daemon is preconfigured with default properties. If your enterprise functions well with default settings, the SLP deployment requires virtually no administration.

In some situations, however, you might want to modify the SLP properties to tune network operations or to activate certain features. With a few configuration changes you can enable SLP logging, for example. The information in a SLP log and in `snoop` traces can then help you decide if additional configuration is necessary.

SLP configuration properties reside in the `slp.conf` file, which is located in the `/etc/inet` directory. If you decide to change the default property settings, refer to Chapter 3, "Administering SLP (Tasks)," for the appropriate procedures.

Before you modify SLP configuration settings, consider the following questions that are related to key aspects of network administration:

- What network technologies are operating in the enterprise?
- How much network traffic can the technologies handle smoothly?
- How many services, of what type, are available on the network?
- How many users are on the network? What services do they require? Where are users located in relation to their most frequently accessed services?

Deciding What to Reconfigure

You can use the SLP-enabled `snoop` utility and SLP logging utilities to decide if reconfiguration is necessary and what properties you need to modify. For example, you might reconfigure certain properties to do the following:

- Accommodate a mix of network media that have varying latencies and bandwidth characteristics
- Recover the enterprise from network failures or unplanned partitioning
- Add DAs to reduce proliferation of SLP multicasts
- Implement new scopes to organize users with their most frequently accessed services

Using `snoop` to Monitor SLP Activity

The `snoop` utility is a passive administrative tool that provides network traffic information. The utility itself generates minimal traffic and enables you to watch all activity on your network as it occurs.

The `snoop` utility provides traces of the actual SLP message traffic. For example, when you run `snoop` with the `slp` command-line argument, the utility displays traces with information on SLP registrations and deregistrations. You can use the information to gauge the network load by checking which services are being registered and how much reregistration activity its occurring.

The `snoop` utility is also useful for observing the traffic flow between SLP hosts in your enterprise. When you run `snoop` with the `slp` command-line argument, you can monitor the following types of SLP activity to determine if network or agent reconfiguration is needed:

- The number of hosts that are using a particular DA. Use this information to decide whether to deploy additional DAs for load balancing.
- The number of hosts that are using a particular DA. Use this information to help you determine whether to configure certain hosts with new or different scopes.
- Whether UA requests a timeout or DA acknowledgment is slow. You can determine whether a DA is overloaded by monitoring UA timeouts and retransmissions. You can also check if the DA requires more than a few seconds to send registration acknowledgments to an SA. Use this information to rebalance the network load on the DA, if necessary, by deploying additional DAs or changing the scope configurations.

Using `snoop` with the `-V` (verbose) command-line argument, you can obtain registration lifetimes and value of the fresh flag in `SrvReg` to determine whether the number of reregistrations should be reduced.

You can also use snoop to trace other kinds of SLP traffic, such as the following:

- Traffic between UA clients and DAs
- Traffic between multicasting UA clients and replying SAs

For more information about snoop, refer to the snoop(1M).

Tip – Use the netstat command in conjunction with snoop to view traffic and congestion statistics. For more information about netstat, refer to netstat(1M).

▼ How to Use snoop to Run SLP Traces

1 Become an administrator.

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 Run snoop with the slp command-line argument.

```
Brief Mode:
# snoop slp
```

When you run snoop in the default *brief* mode, ongoing output is delivered to your screen. SLP messages are truncated to fit on one line per SLP trace.

```
Verbose Mode:
# snoop -v slp
```

When you run snoop in *verbose* mode, snoop delivers ongoing, unabbreviated output to your screen, which provides the following information:

- The complete address of the service URL
- All service attributes
- The registration lifetime
- All security parameters and flags, if any are available

Note – You can use the slp command-line argument with other snoop options.

Analyzing a snoop slp Trace

In the following example, slpd runs on *slphost1* in the default mode as an SA server. The SLP daemon initializes and registers *slphost2* as an echo server. Then, the snoop slp process is invoked on *slphost1*.

Note – To simplify the description of the trace results, the lines in the following snoop output are flagged with line numbers.

```
(1) slphost1 -> 239.255.255.253 SLP V@ SrvRqst [24487] service:directory-agent []
(2) slphost2 -> slphost1 SLP V2 DAAdvert [24487] service:directory-agent://129
(3) slphost1 -> 239.255.255.253 SLP V2 SrvRqst [24487] service:directory-agent []
(4) slphost1 -> 239.255.255.253 SLP V2 SrvRqst [24487] service:directory-agent []
(5) slphost1 -> slphost2 SLP V2 SrvReg [24488/tcp]service:echo.sun:tcp://slphost1:
(6) slphost2 -> slphost1 SLP V2 SrvAck [24488/tcp] ok
(7) slphost1 -> slphost2 SLP V2 SrvDereg [24489/tcp] service:echo.sun:tcp://slphost1:
(8) slphost2 -> slphost1 SLP V2 SrvAck [24489/tcp] ok
```

1. Shows `slpd` on *slphost1* performing active directory agent discovery by multicasting to the SLP multicast group address in search of directory agents. The message number, 24487, for the active discovery is indicated in square brackets in the trace display.
2. Indicates that the active discovery request 24487 from trace 1 is answered by `slpd`, which is running as a DA on the host *slphost2*. The service URL from *slphost2* has been truncated to fit on a single line. The DA has sent a DA advertisement in reply to the multicast directory agent discovery message, as indicated by the matching message numbers in traces 1 and 2.
3. Shows multicasts from the UAs on *slphost1* for additional DAs. Because *slphost2* has already answered the request, it refrains from responding again, and no other DAs reply.
4. Repeats the multicast operation that is shown in the previous line.
5. Shows a `slpd` on *slphost1* forwarding SA client registrations to the DA on *slphost2*. A unicast service registration (`SrvReg`) for an echo server is made by *slphost1* to the DA on *slphost2*.
6. Shows *slphost2* responding to the *slphost1* `SrvReg` with a service acknowledgment (`SrvAck`) that indicates the registration is successful.

   Traffic between the echo server that runs the SA client and the SLP daemon on *slphost1* does not appear in the `snoop` trace. This absence of information is because the `snoop` operation is performed over the network loopback.
7. Shows the echo server on *slphost1* deregistering the echo service advertisement. The SLP daemon on *slphost1* forwards the deregistration to the DA on *slphost2*.
8. Shows *slphost2* responding to the *slphost1* with a service acknowledgment (`SrvAck`) that indicates that the deregistration is successful.

   The `/tcp` parameter that is appended to the message number on lines 5, 6, 7, and 8 indicates that the message exchange occurred by TCP.

Where to Go From Here

After monitoring the SLP traffic, you can use the information that was collected from the `snoop` traces to help determine whether any reconfiguration of the SLP defaults is needed. Use the

related information in Chapter 3, "Administering SLP (Tasks)," for configuring SLP property settings. For more information about SLP messaging and service registrations, refer to Chapter 5, "SLP (Reference)."

◆ ◆ ◆ CHAPTER 3

Administering SLP (Tasks)

The following sections provide information and tasks for configuring SLP agents and processes.

- "Configuring SLP Properties" on page 27
- "Modifying DA Advertising and Discovery Frequency" on page 30
- "Accommodating Different Network Media, Topologies, or Configurations" on page 34
- "Modifying Timeouts on SLP Discovery Requests" on page 39
- "Deploying Scopes" on page 42
- "Deploying DAs" on page 45
- "SLP and Multihoming" on page 49

Configuring SLP Properties

SLP configuration properties control network interactions, SLP agent characteristics, status, and logging. In most situations, the default configuration of these properties requires no modification. However, you can use the procedures in this chapter when the network medium or topology changes and to achieve the following goals:

- Compensate for network latencies
- Reduce congestion on the network
- Add agents or reassign IP addresses
- Activate SLP logging

You can edit the SLP configuration file, `/etc/inet/slp.conf`, to perform operations such as those shown in the following table.

TABLE 3–1 SLP Configuration Operations

Operation	Description
Specify whether `slpd` should act as a DA server. SA server is the default.	Set the `net.slpisDA` property to `True`.

TABLE 3–1 SLP Configuration Operations *(Continued)*

Operation	Description
Set timing for DA multicast messages.	Set the `net.slp.DAHeartBeat` property to control how often a DA multicasts an unsolicited DA advertisement.
Enable DA logging to monitor network traffic.	Set the `net.slp.traceDATraffic` property to `True`.

SLP Configuration File: Basic Elements

The `/etc/inet/slp.conf` file defines and activates all SLP activity each time you restart the SLP daemon. The configuration file consists of the following elements:

- Configuration properties
- Comment lines and notations

Configuration Properties

All of the basic SLP properties, such as `net.slp.isDA` and `net.slp.DAHeartBeat`, are named in the following format.

```
net.slp.<keyword>
```

SLP behavior is defined by the value of a property or a combination of properties in the `slp.conf` file. Properties are structured as key-value pairs in the SLP configuration file. As shown in the following example, a key-value pair consists of a property name and an associated setting.

```
<property name>=<value>
```

The key for each property is the property name. The value sets the numeric (distance or time), true/false state, or string value parameters for the property. Property values consist of one of the following data types:

- True/False setting (Boolean)
- Integers
- List of integers
- Strings
- List of strings

If the value defined is not allowed, the default value for that property name is used. In addition, an error message is logged using syslog.

Comment Lines and Notations

You can add comments to the `slp.conf` file that describe the nature and function of the line. Comment lines are optional in the file, but can be useful for administration.

Note – Settings in the configuration file are case insensitive. For more information, refer to: Guttman, Erik, James Kempf, and Charles Perkins, "Service Templates and service: scheme," RFC 2609 from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2609.txt]

How to Change Your SLP Configuration

Use this procedure to change the property settings in your SLP configuration file. SLP– enabled client or service software also can alter the SLP configuration by using the SLP API. This API is documented in "An API for Service Location," RFC 2614 from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2614.txt]

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Edit the property settings in the `/etc/inet/slp.conf` file as necessary.**

   Refer to "Configuration Properties" on page 28 for general information about the SLP property settings. See the sections that follow this procedure for examples of different scenarios in which you might change the `slp.conf` properties. See `slp.conf(4)`.

5. **Save your changes and close the file.**

6. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Note – The SLP daemon obtains information from the configuration file when you stop or start `slpd`.

Example 3–1 Setting up `slpd` to Operate as a DA Server

You can change the SA server default to enable `slpd` to operate as a DA server by setting the `net.slp.isDA` property to `True` in the `slpd.conf` file.

```
net.slp.isDA=True
```

In each area, various properties control different aspects of the configuration. The following sections describe different scenarios in which you might change the default property settings that are used in SLP configuration.

Modifying DA Advertising and Discovery Frequency

In situations such as the following, you can modify properties that control the timing of DA advertisements and discovery requests.

- When you want the SA or UA to obtain DA configuration information statically from the `net.slp.DAAddresses` property in the `slp.conf` file, you can disable DA discovery.
- When the network is subject to recurrent partitioning, you can change the frequency of passive advertisements and active discovery.
- If UA and SA clients access DAs on the other side of a dial-up connection, you can decrease the DA heartbeat frequency and the active discovery interval to reduce the number of times a dial-up line is activated.
- If network congestion is high, you can limit multicasting.

The procedures in this section explain how to modify the following properties.

TABLE 3–2 DA Advertisement Timing and Discovery Request Properties

Property	Description
`net.slp.passiveDADetection`	Boolean that specifies whether `slpd` listens for unsolicited DA advertisements
`net.slp.DAActiveDiscoveryInterval`	Value that specifies how often `slpd` performs active DA discovery for a new DA
`net.slp.DAHeartBeat`	Value that specifies how often a DA multicasts an unsolicited DA advertisement

Limiting UAs and SAs to Statically Configured DAs

Sometimes you might need to limit UAs and SAs to obtaining DA addresses from the static configuration information in the `slp.conf` file. In the next procedure, you can modify two properties that cause `slpd` to obtain DA information exclusively from the `net.slp.DAAddresses` property.

▼ How to Limit UAs and SAs to Statically Configured DAs

Use the following procedure to change the `net.slp.passiveDADetection` and the `net.slp.DAActiveDiscoveryInterval` properties.

Note – Use this procedure only on hosts that execute UAs and SAs which are restricted to static configurations.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Set the `net.slp.passiveDADetection` property to `False` in the `slp.conf` file to disable passive discovery. This setting causes `slpd` to ignore unsolicited DA advertisements.**

   ```
   net.slp.passiveDADetection=False
   ```

5. **Set the `net.slp.DAActiveDiscoveryInterval` to `-1` to disable initial and periodic active discovery.**

   ```
   net.slp.DAActiveDiscoveryInterval=-1
   ```

6. **Save your changes and close the file.**

7. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Configuring DA Discovery for Dial-up Networks

If the UAs or SAs are separated from the DA by a dial-up network, you can configure DA discovery to reduce or eliminate the number of discovery requests and DA advertisements. Dial-up networks usually incur a charge when activated. Minimizing extraneous calls can reduce the cost of using the dial-up network.

Note – You can disable DA discovery completely with the method that is described in "Limiting UAs and SAs to Statically Configured DAs" on page 30.

▼ How to Configure DA Discovery for Dial-up Networks

You can use the following procedure to reduce unsolicited DA advertisements and active discovery by increasing the DA heartbeat period and the active discovery interval.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Increase the `net.slp.DAHeartbeat` property in the `slpd.conf` file.**

   `net.slp.DAHeartbeat=`*value*

   value A 32-bit integer that sets the number of seconds for the passive DA advertisement heartbeat

   Default Value=10800 seconds (3 hours)

   Range of Values=2000–259200000 seconds

   For example, you can set the DA heartbeat to approximately 18 hours on a host that is executing a DA:

   ```
   net.slp.DAHeartbeat=65535
   ```

5. **Increase the `net.slp.DAActiveDiscoveryInterval` property in the `slpd.conf` file:**

   `net.slp.DAActiveDiscoveryInterval` *value*

   value A 32–bit integer that sets the number of seconds for DA active discovery queries

   Default Value=900 seconds (15 minutes)

   Range of Values=300–10800 seconds

   For example, you can set the DA active discovery interval to 18 hours on a host that is executing a UA and an SA:

   ```
   net.slp.DAActiveDiscoveryInterval=65535
   ```

6. **Save your changes and close the file.**

7. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Configuring the DA Heartbeat for Frequent Partitions

SAs are required to register with all DAs that support their scopes. A DA can appear after `slpd` has performed active discovery. If the DA supports `slpd` scopes, the SLP daemon registers all advertisements on its host with the DA.

One way `slpd` discovers DAs is by the initial unsolicited advertisement a DA sends when it boots. The SLP daemon uses the periodic unsolicited advertisement (the heartbeat) to determine whether a DA is still active. If the heartbeat fails to appear, the daemon removes the DAs the daemon uses and those the daemon offers to UAs.

Finally, when a DA undergoes a controlled shutdown, it transmits a special DA advertisement that informs listening SA services that it will be out of service. The SLP daemon also uses this advertisement to remove inactive DAs from the cache.

If your network is subject to frequent partitions and SAs are long-lived, `slpd` can remove cached DAs during the partitioning if heartbeat advertisements are not received. By decreasing the heartbeat time, you can decrease the delay before a deactivated DA is restored to the cache after the partition is repaired.

▼ How to Configure DA Heartbeat for Frequent Partitions

Use the following procedure to change the `net.slp.DAHeartBeat` property to decrease the DA heartbeat period.

Note – If DA discovery is completely disabled, the `net.slp.DAAddresses` property must be set in `slp.conf` on the hosts that are executing UAs and SAs so that they access the correct DA.

---

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Decrease the `net.slp.DAHeartBeat` value to 1 hour (3600 seconds). By default, the DA heartbeat period is set to 3 hours (10800 seconds).**

   ```
   net.slp.DAHeartBeat=3600
   ```

5 **Save your changes and close the file.**

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Relieving Network Congestion

If network congestion is high, you can limit the amount of multicast activity. If DAs have not already been deployed in the network, deploying DAs can drastically reduce the amount of SLP-related multicast.

However, even after DAs are deployed, multicast is still necessary for DA discovery. You can reduce the amount of multicast necessary for DA discovery by using the method that is described in "How to Configure DA Discovery for Dial-up Networks" on page 31. You can completely eliminate multicast for DA discovery by using the method that is described in "Limiting UAs and SAs to Statically Configured DAs" on page 30.

Accommodating Different Network Media, Topologies, or Configurations

This section describes possible scenarios in which you can change the following properties to tune SLP performance.

TABLE 3–3 SLP Performance Properties

Property	Description
`net.slp.DAAttributes`	The minimum refresh interval that a DA accepts for advertisements.
`net.slp.multicastTTL`	The *time-to-live* value that is specified for multicast packets.
`net.slp.MTU`	The byte size set for network packets. The size includes IP and TCP or UDP headers.
`net.slp.isBroadcastOnly`	The Boolean that is set to indicate if broadcast should be used for DA and non-DA-based service discovery.

Reducing SA Reregistrations

SAs periodically need to refresh their service advertisements before lifetimes expire. If a DA is handling an extremely heavy load from many UAs and SAs, frequent refreshes can cause the DA to become overloaded. If the DA becomes overloaded, UA requests start to time out and are

then dropped. UA request timeouts have many possible causes. Before you assume that DA overload is the problem, use a `snoop` trace to check the lifetimes of service advertisements that are registered with a service registration. If the lifetimes are short and reregistrations are occurring often, the timeouts are probably the result of frequent reregistrations.

Note – A service registration is a *reregistration* if the FRESH flag is not set. See Chapter 5, "SLP (Reference)," for more information on service registration messages.

How to Reduce SA Reregistrations

Use the following procedure to increase the minimum refresh interval for SAs to reduce reregistrations.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Increase the value of the `min-refresh-interval` attribute of the `net.slp.DAAttributes` property.**

   The default minimum reregistration period is zero. The zero default allows SAs to reregister at any point. In the following example, the interval is increased to 3600 seconds (one hour).

   ```
   net.slp.DAAttributes(min-refresh-interval=3600)
   ```

5. **Save your changes and close the file.**

6. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Configuring the Multicast Time-to-Live Property

The multicast time–to-live property (`net.slp.multicastTTL`) determines the range over which a multicast packet is propagated on your intranet. The multicast TTL is configured by setting the `net.slp.multicastTTL` property to an integer between 1 and 255. The default value of the multicast TTL is 255, which means, theoretically, that the packet routing is unrestricted. However, a TTL of 255 causes a multicast packet to penetrate the intranet to the border routers

on the edge of your administrative domain. Correct configuration of multicast on border routers is required to prevent multicast packets from leaking into the Internet's multicast backbone, or to your ISP.

Multicast TTL scoping is similar to standard IP TTL, with the exception that a TTL comparison is made. Each interface on a router that is multicast enabled is assigned a TTL value. When a multicast packet arrives, the router compares the TTL of the packet with the TTL of the interface. If the TTL of the packet is greater than or equal to the TTL of the interface, the packet TTL is reduced by one, as with the standard IP TTL. If the TTL becomes zero, the packet is discarded. When you use TTL scoping for SLP multicasting, your routers must be properly configured to limit packets to a particular subsection of your intranet.

▼ How to Configure the Multicast Time-to-Live Property

Use the following procedure to reset the `net.slp.multicastTTL` property.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Change the `net.slp.multicastTTL` property in the `slpd.conf` file:**

   `net.slp.multicastTTL=`*value*

   value A positive integer less than or equal to 255 that defines the multicast TTL

   Note – You can reduce the range of multicast propagation by reducing the TTL value. If the TTL value is 1, then the packet is restricted to the subnet. If the value is 32, the packet is restricted to the site. Unfortunately, the term *site* is not defined by RFC 1075, where multicast TTLs are discussed. Values above 32 refer to theoretical routing on the Internet and should not be used. Values below 32 can be used to restrict multicast to a set of accessible subnets, if the routers are properly configured with TTLs.

5. **Save your changes and close the file.**

6. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Configuring the Packet Size

The default packet size for SLP is 1400 bytes. The size should be sufficient for most local area networks. For wireless networks or wide area networks, you can reduce the packet size to avoid message fragmentation and reduce network traffic. For local area networks that have larger packets, increasing the packet size can improve performance. You can determine whether the packet size needs to be reduced by checking the minimum packet size for your network. If the network medium has a smaller packet size, you can reduce the `net.slp.MTU` value accordingly.

You can increase the packet size if your network medium has larger packets. However, unless the service advertisements from SAs or queries from UAs frequently overflow the default packet size, you should not have to change the `net.slp.MTU` value. You can use `snoop` to determine whether UA requests often overflow the default packet size and roll over to use TCP rather than UDP.

The `net.slp.MTU` property measures the complete IP packet size, including the link layer header, the IP header, the UDP or TCP header, and the SLP message.

▼ How to Configure the Packet Size

Use the following procedure to change the default packet size by adjusting the `net.slp.MTU` property.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Change the `net.slp.MTU` property in the `slpd.conf` file:**

   `net.slp.MTU=`*value*

   value A 16–bit integer that specifies the network packet size, in bytes

   Default Value=1400

   Range of Values=128–8192

5. **Save your changes and close the file.**

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Configuring Broadcast-Only Routing

SLP is designed to use multicast for service discovery in the absence of DAs and for DA discovery. If your network does not deploy multicast routing, you can configure SLP to use broadcast by setting the `net.slp.isBroadcastOnly` property to `True`.

Unlike multicast, broadcast packets do not propagate across subnets by default. For this reason, service discovery without DAs in a non-multicast network works only on a single subnet. In addition, special considerations are required when deploying DAs and scopes on networks in which broadcast is used. A DA on a multihomed host can bridge service discovery between multiple subnets with multicast disabled. See "DA Placement and Scope Name Assignment" on page 52 for more information on deploying DAs on multihomed hosts.

▼ How to Configure Broadcast-Only Routing

Use the following procedure to change `net.slp.isBroadcastOnly` property to `True`.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Stop `slpd` and all SLP activity on the host.**

```
# svcadm disable network/slp
```

3 **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4 **Change the `net.slp.isBroadcastOnly` property in the `slpd.conf` file to `True`:**

```
net.slp.isBroadcastOnly=True
```

5 **Save your changes and close the file.**

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Modifying Timeouts on SLP Discovery Requests

Two situations might require that you change the timeouts for SLP discovery requests:

- If the SLP agents are separated by multiple subnets, dial-up lines, or other WANs, the network latency can be high enough that the default timeouts are insufficient for a request or registration to be completed. Conversely, if your network is low latency, you can improve performance by decreasing the timeouts.
- If the network is subject to heavy traffic or a high collision rates, the maximum period that SAs and UAs need to wait before sending a message might be insufficient to assure collision-free transactions.

Changing Default Timeouts

High network latency can cause UAs and SAs to time out before a response returns for requests and registrations. Latency can be a problem if a UA is separated from an SA, or if both a UA and an SA are separated from a DA;either by multiple subnets, a dial-up line, or a WAN. You can determine if latency is a problem by checking whether SLP requests are failing because of timeouts on UA and SA requests and registrations. You can also use the `ping` command to measure the actual latency.

The following table lists configuration properties that control timeouts. You can use the procedures in this section to modify these properties.

TABLE 3–4 Time-out Properties

Property	Description

`net.slp.multicastTimeouts`
`net.slp.DADiscoveryTimeouts`
`net.slp.datagramTimeouts`
	The properties that control timeouts for repeated multicast and unicast UDP message transmissions before the transmission is abandoned.

`net.slp.multicastMaximumWait`	The property that controls the maximum amount of time a multicast message is transmitted before it is abandoned.
`net.slp.datagramTimeouts`	The upper bound of a DA timeout that is specified by the sum of values that are listed for this property. A UDP datagram is repeatedly sent to a DA until a response is received or the time-out bound is reached.

If frequent timeouts are occurring during multicast service discovery or DA discovery, increase the `net.slp.multicastMaximumWait` property from the default value of 15000 milliseconds (15 seconds). Increasing the maximum wait period allows more time for requests on high latency

networks to be completed. After you change the `net.slp.multicastMaximumWait`, you should also modify the `net.slp.multicastTimeouts` and `net.slp.DADiscoveryTimeouts`. The sum of the timeout values for these properties equals the `net.slp.multicastMaximumWait` value.

How to Change Default Timeouts

Use the following procedure to change the SLP properties that control timeouts.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Change the `net.slp.multicastMaximumWait` property in the `slpd.conf` file:**

   `net.slp.multicastMaximumWait=`*value*

   value A 32–bit integer that lists the sum of the values that are set for `net.slp.multicastTimeouts` and `net.slp.DADiscoveryTimeouts`

   Default Value=15000 milliseconds (15 seconds)

   Range of Values=1000 to 60000 milliseconds

   For example, if you determine that multicast requests require 20 seconds (20000 milliseconds), you would adjust the values that are listed for `net.slp.multicastTimeouts` and the `net.slp.DADiscoveryTimeouts` properties to equal 20000 milliseconds.

   ```
   net.slp.multicastMaximumWait=20000
   net.slp.multicastTimeouts=2000,5000,6000,7000
   net.slp.DADiscoveryTimeouts=3000,3000,6000,8000
   ```

5. **If necessary, change the `net.slp.datagramTimeouts` property in the `slpd.conf` file:**

   `net.slp.datagramTimeouts=`*value*

   value A list of 32–bit integers that specify timeouts, in milliseconds, to implement unicast datagram transmission to DAs

   Default=3000,3000,3000

For example, you can increase the datagram timeout to 20000 milliseconds to avoid frequent timeouts.

```
net.slp.datagramTimeouts=2000,5000,6000,7000
```

In high-performance networks, you can reduce the time-out bound for multicast and unicast UDP datagram transmission. When you reduce the time-out bound, you decrease latency that is required to satisfy SLP requests.

6 **Save your changes and close the file.**

7 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Configuring the Random-Wait Bound

In networks with heavy traffic or a high collision rate, communication with a DA might be affected. When collision rates are high, the sending agent must retransmit the UDP datagram. You can determine if retransmission is occurring by using `snoop` to monitor traffic on a network of hosts that are running `slpd` as an SA server and a host that is running `slpd` as a DA. If multiple service registration messages for the same service appear in the `snoop` trace from the host that is running `slpd` as an SA server, you might have notice collisions.

Collisions can be particularly troubling at boot time. When a DA first starts, it sends unsolicited advertisements and the SAs respond with registrations. SLP requires the SAs to wait for a random amount of time after receiving a DA advertisement before responding. The random-wait bound is uniformly distributed with a maximum value that is controlled by the `net.slp.randomWaitBound`. The default random-wait bound is 1000 milliseconds (1 second).

▼ How to Configure the Random-Wait Bound

Use the following procedure to change the `net.slp.RandomWaitBound` property in the `slp.conf` file.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Stop `slpd` and all SLP activity on the host.**

```
# svcadm disable network/slp
```

3 **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4 **Change the `net.slp.RandomWaitBound` property in the `slpd.conf` file:**

`net.slp.RandomWaitBound=`*value*

value The upper bound for calculating the random-wait time before attempting to contact a DA

Default Value=1000 milliseconds (1 second)

Range of Values=1000 to 3000 milliseconds

For example, you can lengthen the maximum wait to 2000 milliseconds (2 seconds).

```
net.slp.randomWaitBound=2000
```

When you lengthen the random-wait bound, a longer delay in registration occurs. SAs can complete registrations with newly discovered DAs more slowly to avoid collisions and timeouts.

5 **If necessary, change the `net.slp.datagramTimeouts` property in the `slpd.conf` file:**

`net.slp.datgramTimeouts=`*value*

value A list of 32–bit integers that specify timeouts, in milliseconds, to implement unicast datagram transmission to DAs

Default=3000,3000,3000

For example, you can increase the datagram timeout to 20000 milliseconds to avoid frequent timeouts.

```
net.slp.datagramTimeouts=2000,5000,6000,7000
```

In high-performance networks, you can reduce the time-out bound for multicast and unicast UDP datagram transmission. This setting reduces the amount of latency in satisfying SLP requests.

6 **Save your changes and close the file.**

7 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Deploying Scopes

With scopes, you can provision services that depend on the logical, physical, and administrative groupings of users. You can use scopes to administer access to service advertisements.

Use the `net.slp.useScopes` property to create scopes. For example, in the `/etc/inet/slp.conf` file on a host, add a new scope, called `newscope`, as shown:

```
net.slp.useScopes=newscope
```

Your organization might, for example, have an alcove of networked devices, such as printers and fax machines, at the end of the south hall on the second floor of Building 6. These devices could be used by everyone on the second floor, or you might restrict the usage to members of a certain department. Scopes provide a way for you to provision access to the service advertisements for these machines.

If the devices are dedicated to a single department, you can create a scope with the department name, for example, `mktg`. Devices that belong to other departments can be configured with different scope names.

In another scenario, the departments might be dispersed. For instance, the mechanical engineering and the CAD/CAM departments might be split between floors 1 and 2. However, you can provide the floor 2 machines for the hosts on both floors by assigning them to the same scope. You can deploy scopes in any manner that operates well with your network and users.

Note – UAs that have particular scope are not prevented from actually using services that are advertised in other scopes. Configuring scopes controls only which service advertisements a UA detects. The service is responsible for enforcing any access control restrictions.

When to Configure Scopes

SLP can function adequately without any scope configuration. In the Oracle Solaris operating environment, the default scope for SLP is `default`. If no scopes are configured, `default` is the scope of all SLP messages.

You can configure scopes in any of the following circumstances.

- The organizations you support want to restrict service advertisement access to their own members.
- The physical layout of the organization you support suggests that services in a certain area be accessed by particular users.
- The service advertisements that are appropriate for specific users to see must be partitioned.

An example of the first circumstance was cited in "Configuring DA Discovery for Dial-up Networks" on page 31. An example of the second is a situation in which an organization is spread between two buildings, and you want users in a building to access local services in that building. You can configure users in Building 1 with the `B1` scope, while you configure users in Building 2 with the `B2` scope.

Considerations When Configuring Scopes

When you modify the `net.slp.useScopes` property in the `slpd.conf` file, you configure scopes for all agents on the host. If the host is running any SAs or is acting as a DA, you must configure this property if you want to configure the SAs or DA into scopes other than `default`. If only UAs are running on the machine and the UAs should discover SAs and DAs supporting scopes other than `default`, you do not need to configure the property unless you want to restrict the scopes the UAs use. If the property is not configured, UAs can automatically discover available DAs and scopes through `slpd`. The SLP daemon uses active and passive DA discovery to find DAs, or it uses SA discovery if no DAs are running. Alternatively, if the property is configured, UAs use only the configured scopes and do not discard them.

If you decide to configure scopes, you should consider keeping the `default` scope on the list of configured scopes unless you are sure that all SAs in the network have scopes configured. If any SAs are left unconfigured, UAs with configured scopes are unable to find them. This situation occurs because the unconfigured SAs automatically have scope `default`, but the UAs have the configured scopes.

If you also decide to configure DAs by setting the `net.slp.DAAddresses` property, be sure that the scopes that are supported by the configured DAs are the same as the scopes that you have configured with the `net.slp.useScopes` property. If the scopes are different, `slpd` prints an error message when it is restarted.

▼ How to Configure Scopes

Use the following procedure to add scope names to the `net.slp.useScopes` property in the `slp.conf` file.

1 Become an administrator.

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 Stop `slpd` and all SLP activity on the host.

```
# svcadm disable network/slp
```

3 Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.

4 Change the `net.slp.useScopes` property in the `slpd.conf` file:

`net.slp.useScopes=`<*scope names*>

scope names A list of strings that indicate which scopes a DA or SA is allowed to use when making requests, or which scopes a DA must support

Default Value=Default for SA and DA/Unassigned for UA

Note –

Use the following to construct scope names:

- Any alphanumeric characters, uppercase or lowercase
- Any punctuation characters (except for: ", \, !, <, =, >, and ~)
- Spaces that are considered part of the name
- Non-ASCII characters

  You use a backslash to escape non-ASCII characters. For example, UTF-8 encoding uses `0xc3a9` hex code to represent the letter *e* with the French *aigue* accent. If the platform does not support UTF-8, you use the UTF-8 hex code as the escape sequence `\c3\a9`.

For example, to specify scopes for `eng` and `mktg` groups in `bldg6`, you change the `net.slp.useScopes` line to the following.

```
net.slp.useScopes=eng,mktg,bldg6
```

5 **Save your changes and close the file.**

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Deploying DAs

This section describes the strategic deployment of DAs in a network that is running SLP.

SLP functions adequately with only the base agents (UAs and SAs), and with no deployed DAs or configured scopes. All agents that lack specific configurations use the `default` scope. DAs serve as caches for service advertisements. Deploying DAs decreases the number of messages that are sent on the network and reduces the time that is required to receive responses to messages. This capability enables SLP to accommodate larger networks.

Why Deploy an SLP DA?

The primary reason to deploy DAs is to reduce the amount of multicast traffic and the delays that are associated with gathering unicast replies. In a large network with many UAs and SAs, the amount of multicast traffic that is involved in service discovery can become so large that network performance degrades. By deploying one or more DAs, UAs must unicast to DAs for

service and SAs must register with DAs by using unicast. The only SLP-registered multicast in a network with DAs is for active and passive DA discovery.

SAs register automatically with any DAs they discover within a set of common scopes, rather than accepting multicast service requests. Multicast requests in scopes that are not supported by the DA are still answered directly by the SA, however.

Service requests from UAs are unicast to DAs rather than multicast onto the network when a DA is deployed within the UA's scopes. Consequently, DAs within the UA's scopes reduce multicast. By eliminating multicast for normal UA requests, the time that is required to obtain replies to queries is greatly reduced (from seconds to milliseconds).

DAs act as a focal point for SA and UA activity. Deploying one or several DAs for a collection of scopes provides a centralized point for monitoring SLP activity. By turning on DA logging, it is easier to monitor registrations and requests than by checking the logs from multiple SAs that are scattered around the network. You can deploy any number of DAs for a particular scope or scopes, depending on the need to balance the load.

In networks without multicast routing enabled, you can configure SLP to use broadcast. However, broadcast is very inefficient, because it requires each host to process the message. Broadcast also does not normally propagate across routers. As a result, in a network without multicast routing support, services can be discovered only on the same subnet. Partial support for multicast routing leads to inconsistent ability to discover services on a network. Multicast messages are used to discover DAs. Partial support for multicast routing, therefore, implies that UAs and SAs register services with all known DAs in the SA's scope. For example, if a UA queries a DA that is called `DA1` and the SA has registered services with `DA2`, the UA will fail to discover a service. See "Configuring Broadcast-Only Routing" on page 38 for more information on how to deploy SLP on networks without multicast enabled.

On a network with inconsistent site-wide support for multicast routing, you must configure the SLP UAs and SAs with a consistent list of DA locations by using the `net.slp.DAAdresseses` property.

Finally, the SLPv2 DA supports interoperability with SLPv1. SLPv1 interoperability is enabled by default in the DA. If your network contains SLPv1 devices, such as printers, or you need to interoperate with Novell Netware 5, which uses SLPv1 for service discovery, you should deploy a DA. Without a DA, the Oracle Solaris SLP UAs are unable to find SLPv1 advertised services.

When to Deploy DAs

Deploy DAs on your enterprise if any of the following conditions are true:

- Multicast SLP traffic exceeds 1 percent of the bandwidth on your network, as measured by `snoop`.
- UA clients experience long delays or timeouts during multicast service requests.
- You want to centralize the monitoring of SLP service advertisements for particular scopes on one or several hosts.
- Your network does not have multicast enabled and consists of multiple subnets that must share services.
- Your network employs devices that support earlier versions of SLP (SLPv1) or you would like SLP service discovery to interoperate with Novell Netware 5.

▼ How to Deploy DAs

Use the following procedure to set the `net.slp.isDA` property to `True` in the `slp.conf` file.

Note – You can assign only one DA per host.

---

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Set the `net.slp.isDA` property in the `slpd.conf` file to `True`:**

   ```
   net.slp.isDA=True
   ```

5. **Save your changes and close the file.**

6. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Where to Place DAs

This section provides suggestions for where to place DAs in different situations.

- When multicast routing is not enabled and DAs are required to bridge service discovery between subnets

  In this situation, a DA must be placed on a host with interfaces and all subnets that share services. The `net.slp.interfaces` configuration property does *not* need to be set, unless IP packets are not routed among the interfaces. See "Multihoming Configuration for SLP" on page 49 for more information on configuring the `net.slp.interfaces` property.
- When DAs are deployed for scalability and the primary consideration is optimizing agent access

  UAs typically make many requests for services to DAs. An SA registers with the DA once, and can refresh the advertisement at periodic but infrequent intervals. As a result, UA access to DAs is far more frequent than SA access. The number of service advertisements is also usually smaller than the number of requests. Consequently, most DA deployments are more efficient if the deployment is optimized for UA access.
- Situating DAs so that they are topologically close to UAs on the network to optimize UA access

  Naturally, you must configure the DA with a scope that is shared by both the UA and SA clients.

Placing Multiple DAs for Load Balancing

You can deploy multiple DAs for the same collection of scopes as a means of load balancing. Deploy DAs in any of the following circumstances:

- UA requests to a DA are timing out, or are returning with the `DA_BUSY_NOW` error.
- The DA log shows that many SLP requests are being dropped.
- The network of users who share services in the scopes spans a number of buildings or physical sites.

You can run a `snoop` trace of SLP traffic to determine how many UA requests return with the `DA_BUSY_NOW` error. If the number of UA requests returned is high, UAs in buildings physically and topologically distant from the DA can exhibit slow response or excessive timeouts. In such a scenario, you can deploy a DA in each building to improve response for UA clients within the building.

Links that connect buildings are often slower than the local area networks within the buildings. If your network spans multiple buildings or physical sites, set the `net.slp.DAAddresses` property in the `/etc/inet/slp.conf` file to a list of specific host names or addresses so that the UAs access only the DAs you specify.

If a particular DA is using large amounts of host memory for service registrations, reduce the number of SA registrations by reducing the number of scopes the DA supports. You can split into two a scope that has many registrations. You can then support one of the new scopes by deploying another DA on another host.

SLP and Multihoming

A multihomed server acts as a host on multiple IP subnets. The server can sometimes have more than one network interface card and can act as a router. IP packets, including multicast packets, are routed between the interfaces. In some situations, routing between interfaces is disabled. The following sections describe how to configure SLP for such situations.

Multihoming Configuration for SLP

Without configuration, `slpd` listens for multicast and for UDP/TCP unicast on the default network interface. If unicast and multicast routing is enabled between interfaces on a multihomed machine, no additional configuration is needed. This is because multicast packets that arrive at another interface are properly routed to the default. As a result, multicast requests for DA or other service advertisements arrive at `slpd`. If routing is not turned on for some reason, configuration is required.

When to Configure for Nonrouted, Multiple Network Interfaces

If one of the following conditions exist, you might need to configure multihomed machines.

- Unicast routing is enabled between the interfaces and multicast routing is disabled.
- Unicast routing and multicast routing are both disabled between the interfaces.

When multicast routing is disabled between interfaces, it is usually because multicast has not been deployed in the network. In that situation, broadcast is normally used for service discovery that is not DA-based and for DA discovery on the individual subnets. Broadcast is configured by setting the `net.slp.isBroadcastOnly` property to `True`.

Configuring Nonrouted, Multiple Network Interfaces (Task Map)

TABLE 3–5 Configuring Nonrouted, Multiple Network Interfaces

Task	Description	For Instructions
Configure the `net.slp.interfaces` property	Set this property to enable `slpd` to listen for unicast and multicast/broadcast SLP requests on the specified interfaces.	"Configuring the `net.slp.interfaces` Property" on page 50
Arrange proxy service advertisements so that UAs on subnets get service URLs with reachable addresses	Restrict proxy advertisement to a machine that is running `slpd` connected to a single subnet rather than a multihomed host.	"Proxy Advertising on Multihomed Hosts" on page 52
Place the DAs and configure scopes to assure reachability between UAs and SAs	Configure the `net.slp.interfaces` property on multihomed hosts with a single interface host name or address. Run a DA on a multihomed host, but configure scopes so that SAs and UAs on each subnet use different hosts.	"DA Placement and Scope Name Assignment" on page 52

Configuring the net.slp.interfaces Property

If the `net.slp.interfaces` property is set, `slpd` listens for unicast and multicast/broadcast SLP requests on the interfaces that are listed in the property, rather than on the default interface.

Usually, you set the `net.slp.interfaces` property in conjunction with enabling broadcast by setting the `net.slp.isBroadcastOnly` property, because multicast has not been deployed in the network. However, if multicast has been deployed, but is not being routed on this particular multihomed host, a multicast request can arrive at `slpd` from more than one interface. This situation can occur when the routing of packets is handled by another multihomed host or router that connects the subnets that are served by the interfaces.

When such a situation occurs, the SA server or the UA that is sending the request receives two responses from `slpd` on the multihomed host. The responses are then filtered by the client libraries and the client does not see them. The responses are, however, visible in the `snoop` trace.

Note –

If unicast routing is turned off, services that are advertised by SA clients on multihomed hosts might not be reachable from all the subnets. If services are unreachable, SA clients can do the following:

- Advertise one service URL for each individual subnet.
- Assure that requests from a particular subnet are answered with a reachable URL.

The SA client library makes no effort to assure that reachable URLs are advertised. The service program, which might or might not handle a multihomed host with no routing, is then responsible for assuring that reachable URLs are advertised.

Before you deploy a service on a multihomed host with unicast routing disabled, use `snoop` to determine whether the service handles requests from multiple subnets correctly. Furthermore, if you plan to deploy a DA on the multihomed host, see "DA Placement and Scope Name Assignment" on page 52.

How to Configure the net.slp.interfaces Property

Use the following procedure to change the `net.slp.interfaces` property in the `slp.conf` file.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Change the `net.slp.interfaces` property in the `slpd.conf` file:**

   `net.slp.interfaces=`*value*

   value List of IPv4 addresses or host names of the network interface cards on which the DA or SA should listen for multicast, unicast UDP, and TCP messages on port 427

   For example, a server with three network cards and multicast routing that is turned off is connected to three subnets. The IP addresses of the three network interfaces are `192.147.142.42`, `192.147.143.42`, and `192.147.144.42`. The subnet mask is `255.255.255.0`. The following property setting causes `slpd` to listen on all three interfaces for unicast and multicast/broadcast messaging:

   ```
   net.slp.interfaces=192.147.142.42,192.147.143.42,192.147.144.42
   ```

Note – You can specify IP addresses or resolvable host names for the `net.slp.interfaces` property.

5 **Save your changes and close the file.**

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Proxy Advertising on Multihomed Hosts

If a host with multiple interfaces advertises services by using `slpd` and proxy registration, the service URLs that are advertised by `slpd` must contain reachable host names or addresses. If unicast routing is enabled between the interfaces, hosts on all subnets can reach hosts on other subnets. Proxy registrations can also be made for a service on any subnet. If, however, unicast routing is disabled, service clients on one subnet cannot reach services on another subnet through the multihomed host. However, those clients might be able to reach the services through another router.

For example, suppose the host with default host name `bigguy` has three interface cards on three different unrouted subnets. The host names on these subnets are `bigguy`, with IP address `192.147.142.42`, `bigguy1`, with IP address `192.147.143.42`, and `bigguy2`, with IP address `192.147.144.42`. Now suppose that a legacy printer, `oldprinter`, is connected to the `143` subnet and that the URL `service:printing:lpr://oldprinter/queue1` is configured with the `net.slp.interfaces` to listen on all interfaces. The `oldprinter` URL is proxy-advertised on all interfaces. The machines on the `142` and `144` subnets receive the URL in response to service requests, but are unable to access the `oldprinter` service.

The solution to this problem is to perform the proxy advertisement with `slpd` running on a machine that is connected to the `143` subnet only, rather than on the multihomed host. Only hosts on the `143` subnet can obtain the advertisement in response to a service request.

DA Placement and Scope Name Assignment

The placement of DAs and assignment of scope names on a network with a multihomed host must be done carefully to assure that clients obtain accessible services. Be particularly cautious when routing is disabled and the `net.slp.interfaces` property is configured. Again, if unicast routing is enabled between the interfaces on a multihomed machine, no special DA and scope configuration is necessary. The advertisements are cached with the DA identify services that are accessible from any of the subnets. However, if unicast routing is disabled, poor placement of DAs can result in problems.

To see what problems can result in the previous example, consider what would happen if `bigguy` runs a DA, and clients on all subnets have the same scopes. SAs on the `143` subnet register their service advertisements with the DA. UAs on the `144` subnet can obtain those service advertisements, even though hosts on the `143` subnet are unreachable.

One solution to this problem is to run a DA on each subnet and not on the multihomed host. In this situation, the `net.slp.interfaces` property on the multihomed hosts should be configured with a single interface host name or address, or it should be left unconfigured, forcing the default interface to be used. A disadvantage of this solution is that multihomed hosts are often large machines that could better handle the computational load of a DA.

Another solution is to run a DA on the multihomed host, but configure scopes so that the SAs and UAs on each subnet have a different scope. For example, in the previous situation, UAs and SAs on the `142` subnet might have a scope that is called `scope142`. UAs and SAs on the `143` subnet might have another scope that is called `scope143` and UAs and SAs on the `144` subnet could have third scope that is called `scope144`. You can configure the `net.slp.interfaces` property on `bigguy` with the three interfaces so that the DA serves three scopes on the three subnets.

Considerations When Configuring for Nonrouted, Multiple Network Interfaces

Configuring the `net.slp.interfaces` property enables a DA on the multihomed host to bridge service advertisements between the subnets. Such configuration is useful if multicast routing is turned off in the network, but unicast routing between interfaces on a multihomed host is enabled. Because unicast is routed between the interfaces, hosts on a subnet different from the subnet on which the service is located can contact the service when they receive the service URL. Without the DA, SA servers on a particular subnet receive only broadcasts that were made on the same subnet, so they cannot locate services off of their subnet.

The most common situation that necessitates configuration of the `net.slp.interfaces` property occurs when multicast is not deployed on the network and broadcast is used instead. Other situations require careful thought and planning to avoid unnecessary duplicate responses or unreachable services.

◆ ◆ ◆ CHAPTER 4

Incorporating Legacy Services

Legacy services are network services that predate the development and implementation of SLP. Services such as the the NFS service, and the NIS name service, for example, do not contain internal SAs for SLP. This chapter describes when and how to advertise legacy services.

- "When to Advertise Legacy Services" on page 55
- "Advertising Legacy Services" on page 55
- "Considerations When Advertising Legacy Services" on page 59

When to Advertise Legacy Services

With legacy service advertising, you can enable the SLP UAs to find devices and services such as the following on your networks. You can find hardware devices and software services that do that do not contain SLP SAs. When applications with SLP UAs need to find printers or databases that do not contain SLP SAs, for example, legacy advertising might be required.

Advertising Legacy Services

You use any of the following methods to advertise legacy services.

- Modify the service to incorporate an SLP SA.
- Write a small program that advertises on behalf of a service that is not SLP enabled.
- Use proxy advertising to have `slpd` advertise the service.

Modifying the Service

If the source code for the software server is available, you can incorporate a SLP SA. The C and Java APIs for SLP are relatively straightforward to use. See the man pages for information on the

C API and documentation on the Java API. If the service is a hardware device, the manufacturer might have an updated PROM that incorporates SLP. Contact the device manufacturer for more information.

Advertising a Service That Is Not SLP Enabled

If the source code or an updated PROM that contains SLP is not available, you can write a small application that uses the SLP client library to advertise the service. This application could function as a small daemon that you start or stop from the same shell script you use to start and stop the service.

SLP Proxy Registration

Oracle Solaris `slpd` supports legacy service advertising with a proxy registration file. The proxy registration file is a list of service advertisements in a portable format.

▼ How to Enable SLP Proxy Registration

1 **Create a proxy registration file on the host file system or in any network directory that is accessible by HTTP.**

2 **Determine if a service type template exists for the service.**

The template is a description of the service URL and attributes of a service type. A template is used to define the components of an advertisement for a particular service type:

- If a service type template exists, use the template to construct the proxy registration. See RFC 2609 for more information on service-type templates.
- If a service type template is not available for the service, select a collection of attributes that precisely describe the service. Use a naming authority other than the default for the advertisement. The default naming authority is allowed only for service types that have been standardized. See RFC 2609 for more information on naming authorities.

  For example, suppose a company that is called *BizApp* has a local database that is used to track software defects. To advertise the database, the company might use a URL with the service type `service:bugdb.bizapp`. The naming authority would then be `bizapp`.

3 **Follow the next steps to configure the `net.slp.serializedRegURL` property in the `/etc/inet/slp.conf` file with the location of the registration file that was created in the previous steps.**

4. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

5. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

6. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

7. **Specify the location of the proxy registration file in the `net.slp.serializedRegURL` property of the `/etc/inet/slp.conf` file.**

   `net.slp.net.slp.serializedRegURL=`*proxy registration file URL*

   For example, if the serialized registration file is `/net/inet/slp.reg`, you configure the property as shown in the following:

   ```
   net.slp.serializedRegURL=file:/etc/inet/slp.reg
   ```

8. **Save your changes and close the file.**

9. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

Using SLP Proxy Registration to Advertise

A service advertisement consists of lines that identify the service URL, an optional scope, and a series of attribute definitions. The SLP daemon reads, registers, and maintains proxy advertisements exactly as an SA client would. The following is an example of an advertisement from a proxy registration file.

In the example, a legacy printer that supports LPR protocol and an FTP server are advertised. Line numbers have been added for description purposes and are not part of the file.

```
(1) #Advertise legacy printer.
(2)
(3) service:lpr://bizserver/mainspool,en,65535
(4) scope=eng,corp
(5) make-model=Laserwriter II
(6) location-description=B16-2345
(7) color-supported=monochromatic
(8) fonts-supported=Courier,Times,Helvetica 9 10
(9)
(10) #Advertise FTP server
(11)
(12) ftp://archive/usr/src/public,en,65535,src-server
```

```
(13) content=Source code for projects
(14)
```

Note – The proxy registration file supports the same convention for escaping non-ASCII characters as the configuration file does. For more information about the format of the proxy registration file, see RFC 2614.

TABLE 4–1 SLP Proxy Registration File Description

Line Numbers	Description
1 and 10	Comment lines begin with a cross-hatch symbol (#) and do not affect the file's operation. All characters through the end of a comment line are ignored.
2, 9, and 14	Blank lines that delimit the advertisements.
3, 12	Service URLs that each have three required fields and one optional field that are separated by commas: ■ Generic or `service:` URL advertised. See RFC 2609 for the specification of how to form a service: URL. ■ Language of the advertisement. In the previous example, the field is designated English, *en*. Language is an RFC 1766 language tag. ■ Lifetime of the registration, measured in seconds. The lifetime is restricted to an unsigned 16 bit-integer. If the lifetime is less than the maximum, 65535, `slpd` times out the advertisement. If the lifetime is 65535, `slpd` refreshes the advertisement periodically, and the lifetime is considered permanent, until `slpd` exits. ■ (Optional) Service type field – If used, this field defines the service type. If the service URL is defined, you can change the service type under which the URL is advertised. In the previous example of a proxy registration file, line 12 contains a generic FTP URL. The optional type field causes the URL to be advertised under the service type name *src-server*. The `service` prefix is not added by default to the type name.
4	Scope designation. Optional line consists of the token `scope`, followed by an equal sign and a comma-separated list of scope names. Scope names are defined by the `net.slp.useScopes` configuration property. Only scopes that are configured for the host should be included in the list. When a scope line is not added, the registration is made in all scopes with which `slpd` is configured. The scope line must appear immediately after the URL line. Otherwise, scope names are recognized as attributes.

TABLE 4–1 SLP Proxy Registration File Description *(Continued)*

Line Numbers	Description
5–8	Attribute definitions. After the optional scope line, the bulk of the service advertisement contains attribute/value list pair lines. Each pair consists of the attribute tag, followed by an equal sign, and an attribute value or a comma-separated list of values. In the previous example of a proxy registration file, line 8 illustrates an attribute list with multiple values. All other lists have single values. The format for the attribute names and values is the same as on-the-wire SLP messages.

Considerations When Advertising Legacy Services

Generally, modifying the source code to add SLP is preferable to writing a SLP-enabled service that uses the SLP API to advertise on behalf of other services. Modifying the source code is also preferable to using proxy registration. When you modify the source code, you can add service-specific features and closely track service availability. If the source code is unavailable, writing an SLP-enabled helper service that advertises on behalf of other services is preferable to using proxy registration. Ideally, this helper service is integrated into the service start/stop procedure that is used to control activation and deactivation. Proxy advertising is generally the third choice, when no source code is available and writing a standalone SA is impractical.

Proxy advertisements are maintained only if `slpd` is running to read the proxy registration file. No direct connection exists between the proxy advertisement and the service. If an advertisement times out or `slpd` is halted, the proxy advertisement is no longer available.

If the service is shut down, `slpd` must be stopped. The serialized registration file is edited to comment out or remove the proxy advertisement, and `slpd` is restarted. You must follow the same procedure when the service is restarted or reinstalled. The lack of connection between the proxy advertisement and the service is a major disadvantage of proxy advertisements.

◆ ◆ ◆ CHAPTER 5

SLP (Reference)

This chapter describes the SLP status codes and message types. SLP message types are listed with the abbreviations and function codes. SLP status codes are shown with descriptions and function codes that are used to indicate that a request is received (code 0), or that the receiver is busy.

Note – The SLP daemon (`slpd`) returns status codes for unicast messages only.

SLP Status Codes

TABLE 5–1 SLP Status Codes

Status Type	Status Code	Description
No Error	0	Request was processed without error.
LANGUAGE_NOT_SUPPORTED	1	For an AttrRqst or SrvRqst, there is data for the service type in the scope, but not in the language that is indicated.
PARSE_ERROR	2	The message fails to follow SLP syntax.
INVALID_REGISTRATION	3	The SrvReg has problems. For example, a zero lifetime or an omitted language tag.
SCOPE_NOT_SUPPORTED	4	The SLP message did not include a scope in its scope list that is supported by the SA or DA that answered the request.
AUTHENTICATION_UNKNOWN	5	The DA or SA received a request for an unsupported SLP SPI.
AUTHENTICATION_ABSENT	6	The UA or DA expected URL and attribute authentication in the SrvReg and did not receive it.

TABLE 5–1 SLP Status Codes *(Continued)*

Status Type	Status Code	Description
AUTHENTICATION_FAILED	7	The UA or DA detected an authentication error in an Authentication block.
VER_NOT_SUPPORTED	9	Unsupported version number in message.
INTERNAL_ERROR	10	An unknown error occurred in the DA or SA. For example, the operating system had no remaining file space.
DA_BUSY_NOW	11	The UA or SA should retry, using exponential backoff. The DA is busy processing other messages.
OPTION_NOT_UNDERSTOOD	12	The DA or SA received an unknown option from the mandatory range.
INVALID_UPDATE	13	The DA received a SrvReg without FRESH set, for an unregistered service or with inconsistent service types.
MSG_NOT_SUPPORTED	14	The SA received an AttrRqst or SrvTypeRqst and does not support it.
REFRESH_REJECTED	15	The SA sent a SrvReg or partial SrvDereg to a DA more frequently than the DA's min-refresh-interval.

SLP Message Types

TABLE 5–2 SLP Message Types

Message Type	Abbreviation	Function Code	Description
Service Request	`SrvRqst`	1	Issued by a UA to find services or by a UA or SA server during active DA discovery.
Service Reply	`SrvRply`	2	The DA or SA response to a service request.
Service Registration	`SrvReg`	3	Enables SAs to register new advertisements, to update existing advertisements with new and changed attributes, and to refresh URL lifetimes.
Service Deregistration	`SrvDereg`	4	Used by the SA to deregister its advertisements when the service they represent is no longer available.
Acknowledgment	`SrvAck`	5	The DA response to an SA's service request or service deregistration message.
Attribute Request	`AttrRqst`	6	Made either by URL or by service type to request a list of attributes.

TABLE 5–2 SLP Message Types *(Continued)*

Message Type	Abbreviation	Function Code	Description
Attribute Reply	`AttrRply`	7	Used to return the list of attributes.
DA Advertisement	`DAAdvert`	8	The DA response to multicast service requests.
Service Type Request	`SrvTypeRqst`	9	Used to inquire about registered service types that have a particular naming authority and are in a particular set of scopes.
Service Type Reply	`SrvTypeRply`	10	The message that is returned in response to the service type request.
SA Advertisement	`SAAdvert`	11	UAs employ the `SAAdvert` to discover SAs and their scopes in networks where no DAs are deployed.

Index

B

broadcast (SLP), 38, 46, 49

D

DA_BUSY_NOW, 48
DA discovery (SLP), 39
DA heartbeat, frequency, 30
DAs (SLP)
  advertising, 30, 31, 33
  DA logging, 46
  deploying, 34, 45–46
  dial-up networks discovery, 31, 33
  disable active discovery, 31
  disable passive discovery, 31
  discovery, 30, 34, 44
  eliminating multicast, 31
  heartbeat, 33, 35
  multicast, 34
  multiple DAs, 48–49
  removing, 33
  without multicast, 49
directory agent (SLP)
  DA addresses, 30
  load balancing, 48–49
  network congestion and, 34
  SLP architecture and, 15
  when to deploy, 47
  where to place, 48–49
discovery requests (SLP), 39

E

`/etc/inet/slp.conf` file
  broadcast-only routing, 38
  changing configuration, 29
  changing interfaces, 51
  DA advertisments, 32
  DA heartbeat, 33
  deploy DAs, 47
  elements, 28
  load balancing, 48
  multicast time-to-live, 36
  new scopes, 42, 44
  overview, 21
  packet size, 37
  proxy registration, 57
  random-wait bound, 41
  SA reregistrations, 35
  timeouts, 40
  with static DAs, 31
`/etc/init.d/slpd` script, 57

L

legacy services (SLP)
  advertising, 55, 59
  definition, 55
`libslp.so` library, 18

M

message types, SLP, 62–63
multicast (SLP)
  changing interfaces, 50
  DAs, 31, 33
  if disabled, 49
  multihomed machines and, 49
  propagation, 36
  service requests, 46
  time–to-live property, 35
  traffic, 45
multihomed hosts (SLP)
  broadcast-only routing, 38
  changing interfaces, 50
  configuration, 49
  proxy advertisement, 52
  scopes and, 52
  unicast routing disabled, 51
  without multicast, 46

N

net.slp.DAActiveDiscoveryInterval property, 31
  definition, 30
net.slp.DAAddresses property, 33, 44, 48
  definition, 30
net.slp.DAAttributes property, 35
net.slp.DAHeartBeat property, 33, 35
  definition, 30
net.slp.interfaces property
  changing interfaces, 52
  configuring, 50
  DAs and, 48
  multihomed hosts and, 53
  nonrouted interfaces and, 53
net.slp.isBroadcastOnly property, 38, 49, 50
net.slp.isDA property, 29
net.slp.MTU property, 37
net.slp.multicastTTL property, 35
net.slp.passiveDADetection property, 31
  definition, 30
net.slp.randomWaitBound property, 41
net.slp.serializedRegURL property, 56
net.slp.useScopes property, 44, 58
net.slp.useScopes property *(Continued)*
  definition, 42
`netstat` command, 23
network interfaces (SLP), nonrouted
  considerations, 53

P

packet size, configuring for SLP, 37
`ping` command, 39
proxy advertisement (SLP), 55, 57
proxy registration (SLP), 56, 58
  multihomed hosts, 52

R

registration lifetime (SLP), 23

S

SA server (SLP), 41
SAs (SLP), 44, 51, 56
scopes (SLP)
  considerations, 44
  DAs and, 33, 46
  `default` scope, 44
  definition, 15
  deploying, 42–45
  multihomed hosts and, 52
  proxy registration and, 56
  when to configure, 43
service advertisement (SLP), 34, 57
service agent (SLP), 30, 34
service discovery (SLP), 38, 39, 45
service requests (SLP), 46
service URLs
  proxy registration (SLP), 56, 58
SLP
  advertising, 46
  agents and processes, 16–18
  analyzing `snoop` slp trace, 23
  architecture, 15

SLP *(Continued)*
broadcast routing, 38
configuration file, 27, 28–29
configuration properties, 28
configuring, 21–22
daemon, 18
discovery requests, 39
implementation, 18
logging, 15
packet size, 37
performance tuning, 34
planning deployment, 21–22
`slp.conf` file, comments, 28
`slp.jar` library, 18
SLP message types, 62–63
SLP status codes, 61–62
`slpd.conf` file, 30, 44
`slpd` daemon, 55, 56, 59
changing interfaces, 50
DAs, 41
heartbeat, 33
multihomed machines and, 49
proxy advertisement and, 52
removing DAs, 33
SA server, 41
scopes and, 44
static DAs and, 30
SLPv2, interoperability with SLPv1, 46
`snoop` command
monitoring retransmission, 41
multiple SLP requests and, 50
SLP service registration and, 34
SLP traffic and, 48
using with SLP, 22, 23
status codes, SLP, 61–62

T

timeouts (SLP), 39, 46
tuning SLP performance, 34

U

UAs, requests, 34
UAs (SLP), 22, 46
requests timeout, 48
UDP/TCP unicast (SLP), 49
unicast routing (SLP), 49
disabled, 51
user agent (SLP), 30