



An Introduction to Shepherd Directory Services

J&J Computer Consulting Shepherd Server Publications

Overview

This document introduces and describes Shepherd Directory Services (SDS) technology and its capabilities. Understanding SDS requires you to first understand the basics of directory technology, and then to understand how that technology is implemented in Shepherd.

Directory Services

Directories serve as databases of information on users and resources in computing environments. In a lot of cases, directories are little more than fancy yellow pages listing names, addresses, and phone numbers broken into a geographical hierarchy. SDS, however, takes the directory paradigm further and applies it to every part of the computing environment. SDS allows system administrators to maintain a single source of network configuration information across multiple servers and tcp/ip services.

SDS works much like LDAP (lightweight directory access protocol) or X.500 in that it groups objects into a logical hierarchy with multiple levels of countries, organizations, organizational units, localities, groups, users, and network resources. Though SDS is not currently capable of providing information to LDAP or X.500, Shepherd is capable of storing and retrieving its directory data in an LDAP server. Using an LDAP gateway, it should also be capable of working with X.500 servers as well.

Directory Schema

The directory schema is the set of objects and attributes used by a directory to store its information and the rules it uses to act on those objects and attributes. Unfortunately, outside of the basic objects such as users, groups, countries, organizations, etc., there is no standard directory schema. Consequently, making directory services interact and share information is at best difficult, at worst impossible.

SDS supports the following types of objects:

- **[Root] object**
- **Container objects**
- **Leaf objects**

Using these objects, network resources can be laid out in physical or logical formats or a combination of both.

[Root]

The [Root] object does not currently exist in SDS. Due to some questions about the emerging function of LDAP v3.0, SDS currently only operates off of an indicated base to the directory tree. In most cases, this will not affect an organization's implementation plans as an organization entry can be used as the base with organizational unit, organization, and locality objects residing below it to form the directory.

Container Objects

There are four container objects:

- **Country**

Country objects may only reside below the root. For multi-national corporations, multiple country entries allow the organization to be grouped in a logical, geographic way. A country is not, however, required as part of the directory.

- **Locality**

Locality objects can be placed in the [Root], Organization, or Organizational Unit containers. Using locality objects allows the administrator to provide additional geographic grouping in the directory. Locality objects can also hold Organization or Organizational Unit objects.

- **Organization**

Organization objects are typically used to designate a company, university, or government body. These objects can contain Organizational Unit objects or leaf objects.

- **Organizational Unit**

The Organizational Unit objects allow further distinction of parts of your organization. Organizational Unit objects must reside below an Organization object and can contain leaf objects or other Organizational Unit objects.

Leaf Objects

Leaf objects represent network resources such as users, services, servers, etc. They can reside in any container object.

Object Attributes

A directory object consists of a distinguished name (dn) along with several required and optional attributes. Attributes hold the information on an entry like description, objectclass, access control, etc. Attributes may be added to SDS and its objects, but SDS does provide a base set of attributes to enable all of its core services. Table 1 lists the attributes available in SDS and describes their content and purpose.

Table 1: SDS Attributes

Attribute	Contents	Purpose
o	Name of an organization.	Used in the organization object to identify the name of the organization.
ou	Name of the organizational unit.	Used in the organizational unit object to identify a name for the organizational unit.
c	A country identifier like US or CZ.	Represents the country in country objects.
cn	Common name of an entry. For all but container objects, this is the final component of the distinguished name.	Identifies the common name of objects in the directory.
description	A short description of the object in the directory.	Informational
objectclass	The name of one of the object classes used in the directory.	This attribute is used to identify the class of the object. Objectclass is a required attribute for every entry in the directory.
acl	<p data-bbox="431 926 932 957"><access>:<user>:<group>:<ip>:<service></p> <p data-bbox="431 995 740 1026"><access> = CDBPRWXA</p> <p data-bbox="431 1031 553 1062">C - Create</p> <p data-bbox="431 1066 553 1098">D - Delete</p> <p data-bbox="431 1102 570 1134">B - Browse</p> <p data-bbox="431 1138 618 1169">P - Permissions</p> <p data-bbox="431 1173 537 1205">R - Read</p> <p data-bbox="431 1209 553 1241">W - Write</p> <p data-bbox="431 1245 586 1276">X - Execute</p> <p data-bbox="431 1281 570 1312">A - Admin</p> <p data-bbox="431 1358 984 1430"><user> = Name of the user given access. * can be used as a wildcard.</p> <p data-bbox="431 1476 992 1547"><group> = Name of the group given access. * can be used as a wildcard.</p> <p data-bbox="431 1593 951 1694"><ip> = The client ip address from which this user or group can have the specified access. * can be used as a wildcard.</p> <p data-bbox="431 1740 980 1850"><service> = Name of the service through which the identified access can be granted. * can be used as a wildcard.</p>	<p data-bbox="1016 926 1477 1062">The acl attribute sets up the SDS access control mechanism. Further information on SDS access control can be found in the next section.</p>

Attribute	Contents	Purpose
l	Locality	Used to group entries geographically by region, state, or city.
userPassword	Password	The password required for logging onto an SDS system.
emailAddress		
member	The distinguished name of a user that belongs in the group.	If a group is identified in an acl, the group object is found in the directory and the user requesting access is checked against the member attribute.
domainName	An Internet domain name.	
domainNameAlias	A list of aliases for a particular domain.	
ipHostNumber	A 32-bit, dotted decimal Internet address.	Specifies the address of a domain or the address on which a particular service should run.
ipServicePort	A 16-bit port number.	Specifies the port on which a Shepherd Service should run.
faxNumber		
fullName		
generationalQualifier	Jr., Sr., etc.	
initials		
postalAddress		
postalCode		
poBox		
surname	User's last name	
telephoneNumber		
title		
serviceModule	Name of a Shepherd Service module.	Shepherd Service modules reside in initialization files named with a .svc extension. The serviceModule provides information on the DLL and API function names needed by Shepherd to start a service.
servicePriorityClass	* Currently OS/2 specific 1 - Idle 2 - Regular 3 - Time critical 4 - Foreground Server	SDS will start services at a specified priority. This attribute along with servicePriorityDelta sets the exact priority.
servicePriorityDelta	-31 to +31	

Attribute	Contents	Purpose
hostServer	Distinguished Name of a Shepherd Server in the directory.	Used by services and domains to identify which Shepherd Server should handle their requests.
hostDomain	Distinguished Name of a Domain object in the directory.	Used by service specific objects to specify which domain they belong to.
irf	Rights filter. Can be set to a combination of any of the acl access rights except for Admin.	The access control mechanism within SDS allows rights set at a higher level in the directory tree to be filtered out.
serviceLog	Distinguished name of ServiceLog objects in the directory.	Points a service to its ServiceLog objects.

Object Classes

SDS provides several object classes that provide the basis for access control, server initialization, service loading, etc. Table 2 lists each of these object classes with required and optional attributes as well as the purpose.

Table 2: SDS Object Classes

Object Class	Required Attributes	Optional Attributes	Purpose
Country	objectclass c	description acl irf	A container object that represents a country.
Group	objectclass cn member	description acl irf	Represents a group of users. The member object can contain multiple user names.
internetDomain	objectclass cn ipHostNumber hostServer	description acl irf domainNameAlias	Represents an Internet domain name. internetDomain objects are used to assist services in providing virtual hosting.
ShepherdServer	objectclass cn	description acl irf	Represents an actual Shepherd Server. This object is basically a place holder that allows services to link back to it using the hostServer attribute.
Organization	objectclass o	description acl irf	Represents an organization.
OrganizationalUnit	objectclass ou	description acl irf	Represents an organizational unit.

Object Class	Required Attributes	Optional Attributes	Purpose
ShepherdAccount	objectclass cn userPassword	description acl irf	Identifies a user object as one that acts as an SDS user account.
ShepherdService	objectclass cn ipServicePort serviceModule servicePriorityClass servicePriorityDelta hostServer ipHostNumber	description irf acl serviceLog	<p>Gives the Shepherd Server enough information to load the service. ipHostNumber can be an asterisk (*) so that the service starts on all server IP addresses.</p> <p>The serviceLog entry can have multiple entries for any number of log files.</p>
ServiceLog	objectclass cn filename	description irf acl	Represents a log file. Services list each serviceLog that they use in their serviceLog attribute. This object then provides the information to the Shepherd Server for where to find and initialize the log.
Person	objectclass cn	description irf acl emailAddress faxNumber fullName generationalQualifier initials postalAddress postalCode poBox surname telephoneNumber title	Represents a system user. The attributes provided here are insufficient to authenticate a user to the directory. Person and ShepherdAccount objects should be combined to represent a user in an organization.

Shepherd Directory Services Access Control

Rights in an SDS system flow down from the [Root] or base of the directory tree to the leaf objects. The acl attribute of each object sets the rights for that object. In the case of container objects, the acl attribute sets the rights for the container unless they are overridden by a rights filter, or irf. By applying a rights filter at a lower level in the directory tree, rights coming from higher in the tree can be eliminated with the exception of the Admin right which cannot be filtered.

The following access rights are available in SDS:

- **Create**
- **Delete**
- **Browse**
- **Permissions**
- **Read**
- **Write**
- **Execute**
- **Admin**

Each of these access rights can be applied by any service running on an SDS system as well as the local administration console. In the case of normal services, the acl should contain the distinguished name of the service in the acl. For the local administration console, a special service "console" is used for access control. An asterick (*) can be used to represent all services.

In the case of the administration console, these rights relate to directory object access control, but other services will treat these differently while trying to maintain some level of consistency to their application. For instance, the Shepherd.Chat Service uses only read and write access on its room objects to determine if chat users can read messages posted to rooms or write messages and post them to rooms. While the concept of read and write is the same, it is applied not to the directory objects themselves but to the entity they represent. And, Shepherd.Chat does not use any of the other rights with the exception of Admin. Admin will automatically give read and write access to any user that has the Admin right, so it is indirectly used by Shepherd.Chat.

As with other systems, rights can be controlled by a combination of user and group. Group objects in the directory can be referenced by distinguished name in the acl causing the system to look at the member attribute of the Group object. If the user requesting access is found in the Group, access is granted. An asterick (*) can be used to represent all groups. This is typically used when the entry will be for a user as opposed to a group.

The final method of control in the acl is the IP address. The IP address of the acl lets the administrator specify what IP address a particular user can use to logon and receive the indicated access rights. If the IP address does not match, access will not be granted. Again, an asterick(*) can be used to represent all IP addresses. At this time, matching partial IP addresses is not permitted.

With a well organized directory structure, access control in SDS can be both powerful and efficient.

Directory Naming

All objects in the directory are represented by what is referred to as a distinguished name (dn). The dn of any entry specifies the full name back to the root of the directory tree. For instance, the user bjensen at ABC Corp. might have the dn:

```
cn=bjensen, o=ABC Corp., c=US
```

SDS follows LDAP naming rules as closely as possible, but it does deviate in some places. Commas are not currently allowed in distinguished names as SDS will assume they are the break between two components of a name as opposed to being part of a name. Additionally, in the case of attributes that require dn entries, dn's are expected to have one space after each comma and no spaces between attribute name, equal sign, and attribute value. When accessing directory objects, these issues should be considered carefully as objects will not be found in the directory for which an improperly formatted name is given.

Relative names can also be used to reference objects in the directory. Much like an operating system's command line interface, SDS keeps track of the current "context" or "path" in the directory. If the current path is known to the user, a relative name, offset from the current path, can be used to identify an object. For instance, in the bjensen example above, if the current path is o=ABC Corp., c=US, bjensen can be referred to as cn=bjensen.