

## Object Class Definitions

Overview .....	4-3
Components of Object Class Definitions .....	4-3
Structure Rules .....	4-3
Object Class Flags .....	4-9
Default ACL Templates .....	4-12
Super Classes .....	4-12
Object Class Attributes .....	4-14
Reading the Object Class Definitions .....	4-14
Class Flags .....	4-14
Default ACL Template .....	4-15
Subclass of .....	4-15
Containment .....	4-15
Named By .....	4-15
Attributes .....	4-15
Remarks .....	4-15
Individual Object Class Definitions .....	4-15
AFP Server .....	4-16
Alias .....	4-17
Bindery Object .....	4-18
Bindery Queue .....	4-19
CommExec .....	4-21
Computer .....	4-22
Country .....	4-24
Device .....	4-25
Directory Map .....	4-26
External Entity .....	4-27
Group .....	4-28
List .....	4-30
Locality .....	4-31
Message Routing Group .....	4-32
Messaging Server .....	4-33
NCP Server .....	4-35
Organization .....	4-37
Organizational Person .....	4-38
Organizational Role .....	4-39
Organizational Unit .....	4-40
Partition .....	4-41
Person .....	4-42
Print Server .....	4-43

Printer .....	4-45
Profile .....	4-47
Queue .....	4-48
Resource .....	4-50
Server .....	4-51
Top .....	4-53
Tree Root .....	4-54
Unknown .....	4-55
User .....	4-56
Volume .....	4-58

## Overview

In NDS™, object classes serve as the principal templates for storing information in the Directory. All Directory objects must belong to an object class. This chapter describes the object class definitions used to create objects in the Directory tree and how they function in the Directory.

## Components of Object Class Definitions

Object class components define the types of objects that can exist in the Directory tree. In the base schema all object classes are “Nonremovable”; that is, they cannot be deleted or in any other way removed from the schema as defined in the Directory. An object class is defined by its features, which consists of the following kinds of information:

- Structure rules
- Object class flags
- ACL templates
- Super classes
- Object class attributes

### Structure Rules

All object classes possess structure rules that determine how objects of the class are named and where in the Directory tree hierarchy they can be placed. The structure rules for an object class define the possible structural relationships of objects in the Directory tree. The structure rules are either explicitly defined by the class or inherited from a super class. *Naming Attributes* and *Containment Classes* define potential relationships of objects. The naming attributes determine how the object is named. The containment classes determine where the object may appear in the Directory tree relative to other object classes. The *Containment Classes* and *Naming Attributes* of a class comprise a set of structure rules that define an object’s relation to other objects in the Directory.

### Naming Attributes

Objects are identified by their own name and the name of objects they are contained in. The container objects are also called parent objects. An object’s name is referred to as its partial name or *Relative Distinguished Name* (RDN). An object’s RDN is determined by its *Named By* attributes.

The object’s full name (with all its parent names included) is called the complete name or *Distinguished Name* (DN). An object’s DN is determined by all the objects it is subordinate to, but only the immediate parent’s object class must appear in the object’s containment class list. Hence, structure rules effectively control the formation of Distinguished Names.

Each class has one or more attributes designated as naming attributes. The names of the attribute types used for naming are kept in the *Named By* portion

of the object description. These attributes can be either mandatory or optional attributes, but at least one must be given a value when creating an object of that class. If the only *Named By* attribute is declared as optional, it is, in effect, mandatory.

These attributes may be multi-valued and must follow the inheritance rules. For example, objects classed as Organization objects are named using the *O (Organizational Name)* naming attribute. This is the only attribute value that can be used for an organization's partial name. Some object class definitions specify more than one attribute value in naming the object. For example, the Locality object class is named by the *L (Locality Name)* and *S (State or Province Name)* naming attributes. Thus, an RDN for locality might be "L=Provo+S=Utah."

The parts of the name are the locality and the state. The value of the name is "Provo+Utah". When the type specifiers (in this case L and S) are used as shown, it is referred to as a *typed name*.

A naming attribute does not necessarily reflect the class an object belongs to. Many classes, such as *Computer*, *User*, and *Server*, are named by their *CN (Common Name)* naming attribute. In such names, the naming attribute itself does not indicate which class the object belongs to. However, the value of the naming attribute may suggest the nature of the object. On the other hand, some naming attributes are closely tied to specific classes. For example, the *C (Country Name)* naming attribute is used to name *Country* objects.

### **Containment Classes**

For each object class, a multi-valued list of containment classes specifies where an object of that class may appear in the hierarchical structure of the Directory tree. Objects of the class may be created as subordinates in the Directory tree to objects of the classes listed here. An object can be immediately subordinate to only those objects whose classes appear in the containment list of the object's expanded class definition. Containment classes limit the possible location of an object in the Directory tree, thus restricting the order and types of partial names that appear in the object's complete name.

Thus, the containment rules control the Directory tree structure through inheritance properties. Containment helps to ensure that the Directory tree expands in a consistent and logical fashion. For example, a *Country* object can only be subordinate to the root of the tree. Thus, it is always the most significant component where the name of a country is in a complete name. Other objects, however, can also be subordinate to the root, so a country name is not necessarily the most significant component of a complete name.

While helping to control the structure of the Directory, containment classes must also be flexible enough to accommodate a variety of organizational situations. For example, in the relationship between the classes *Organization*

and *Locality*, each class specifies the other as a containment class. This allows an administrator to decide which hierarchical order best represents his organization.

Some object classes can contain other objects, and some object classes cannot. Object classes that can contain other objects are called *Containment Classes*. If an object class is a containment class, the *Container Class Flag* is turned On. Object classes that cannot contain other classes are called *Noncontainer Classes* or *Leaf Classes*, in which case the *Container Class Flag* is turned Off.

The following list of containment classes and leaf classes do not include noneffective classes because they cannot be used to create objects in the Directory. They are only used to define rules for other object classes to inherit. The following classes are containment classes, that is, they can contain other object classes:

- Top
- Tree Root
- Country
- Locality
- Organization
- Organizational Unit

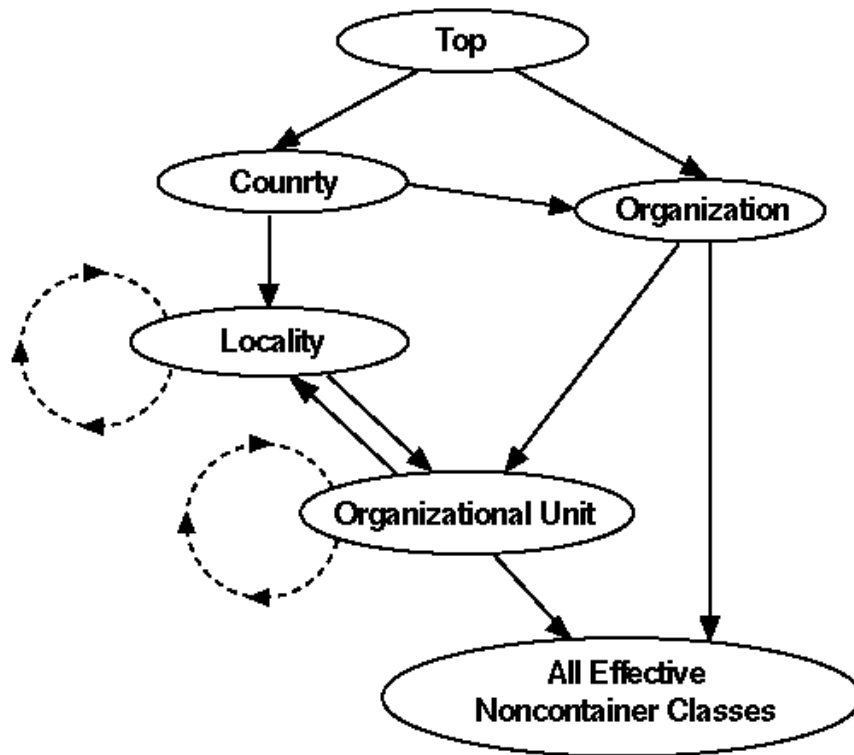
Table 4.1 shows the containment classes and the object types that they can contain.

**Table 4.1**  
**Containment Classes**

<b>Object Class</b>	<b>Contained Classes</b>
Top	Country Organization Tree Root
Tree Root	Country Organization
Country	Locality Organization
Locality	Locality Organization Organizational Unit
Organization	Locality Organizational Unit Leaf Objects
Organizational Unit	Locality Organizational Unit Leaf Objects

Figure 4.1 presents a graphic view of the Directory tree containment structure. This view shows the containment classes and the object classes that they can contain and that can be contained by them. Those object classes that cannot be contained by any of the containment classes are, of course, not shown in this view. The object class *Tree Root* is not shown in this graphical view because it is a virtual object class occupying the same level as *Top*, but inheriting rules and attributes from *Top*.

**Figure 4.1**  
**Directory Tree Containment Structure**



The following classes are noncontainer classes and can be found only as leaf classes. In the Directory tree, a leaf class implies that it is an effective class:

- AFP Server
- Alias
- Bindery Object
- Bindery Queue
- Comm Exec
- Computer
- Directory Map
- External Entity
- Group

- List
- Message Routing Group
- Messaging Server
- NCP Server
- Organizational Role
- Print Server
- Printer
- Profile
- Queue
- Unknown
- User
- Volume

Table 4.2 shows all object classes and their containment status, that is, the object types they can be contained by.

**Table 4.2**  
**Contained By Classes**

<b>Object Class</b>	<b>Classes Contained By</b>
AFP Server	Inherited from Server
Alias	Special case - inherited from Alias object
Bindery Object	Organization Organizational Unit
Bindery Queue	Inherited from Queue
Comm Exec	Inherited from Server
Computer	Inherited from Device
Country	Top Tree Root
Device	Organization Organizational Unit
Directory Map	Inherited from Resource
External Entity	Organization Organizational Unit
Group	Organization Organizational Unit
List	Organization Organizational Unit
Locality	Country Locality Organization Organizational Unit
Message Routing Group	Inherited from Group
Messaging Server	Inherited from Server

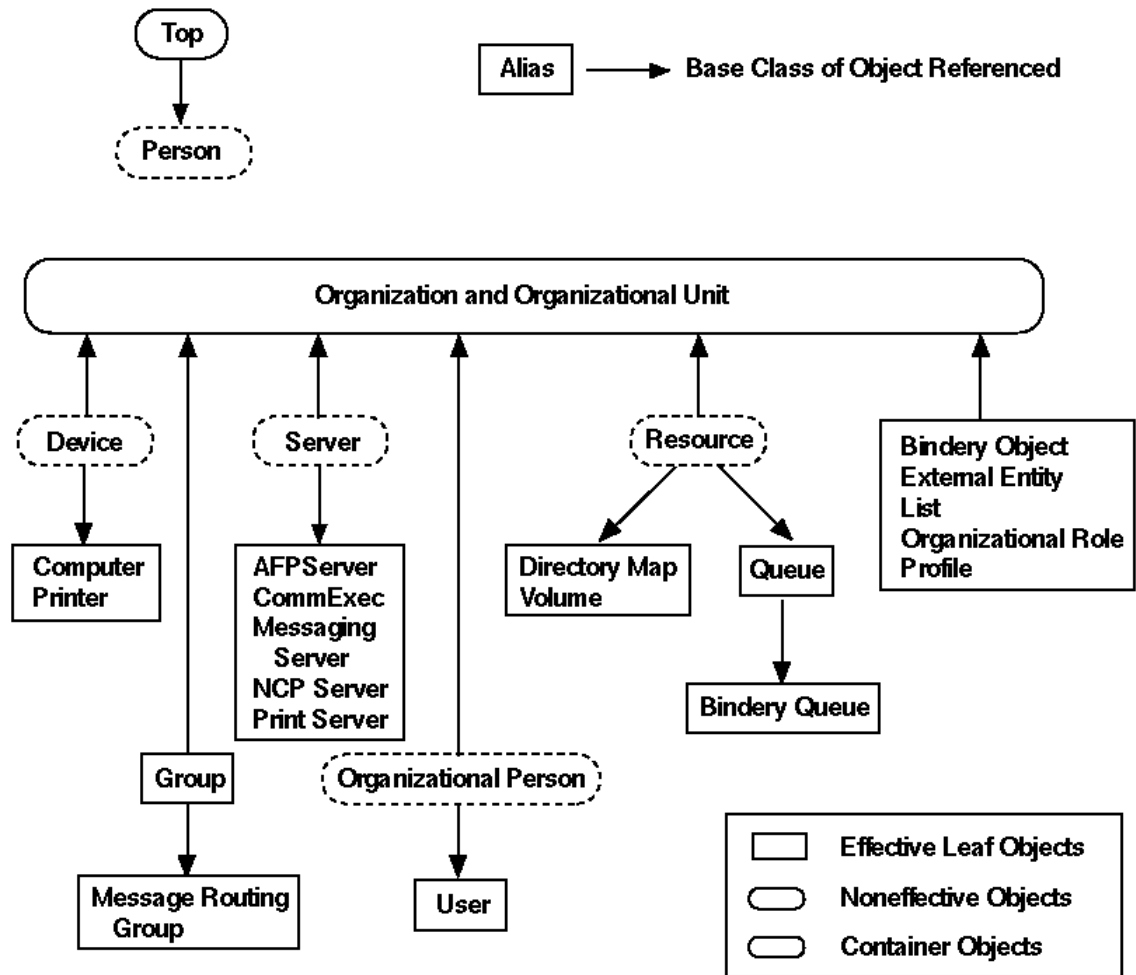
**Table 4.2**  
**Contained By Classes** *continued*

<b>Object Class</b>	<b>Classes Contained By</b>
NCP Server	Inherited from Server
Organization	Top Tree Root Country Locality
Organizational Person	Organization Organizational Unit
Organizational Role	Organization Organizational Unit
Organizational Unit	Locality Organization Organizational Unit
Partition	Special case - not a separate class of object
Person	Inherited from Top
Print Server	Inherited from Server
Printer	Inherited from Device
Profile	Organization Organizational Unit
Queue	Inherited from Resource
Resource	Organization Organizational Unit
Server	Organization Organizational Unit
Top	Special case - root of tree
Tree Root	Inherited from Top
Unknown	Special case - any containment
User	Inherited from Organizational Person
Volume	Inherited from Resource

Figure 4.2 presents a graphical view of how the leaf objects inherited their containment classes. The arrows pointing up to container objects indicate which object class declared the containment classes. Arrows pointing down to a leaf object indicate the objects that inherit the containment classes.



Figure 4.2  
Container Inheritance for Leaf Objects



Two object classes are unique: Person and Alias. They are shown at the top of the graphic. Alias inherits its containment classes from the object that it references. Since all leaf objects have Organization and Organizational Unit as their containment classes, an Alias will usually inherit these containment classes. However, an Alias can reference a container and when it does, the Alias inherits the container's containment classes.

Person is a noneffective class and is the only noneffective class to inherit Top as its containment class.

### Object Class Flags

There are five *Object Class Flags* that may be “set” (turned *On*) or “not set” (turned *Off*): *Container*, *Effective Status*, *Nonremovable*, *Ambiguous Naming*, and *Ambiguous Container*.

### **Container Flag**

The *Container Flag* is “set” (turned *On*) only for those object classes that are designated as container classes. For all leaf object classes the container flag is “not set” (turned *Off*).

### **Effective Status Flag**

Object classes can be effective or noneffective. Only effective classes can be used to create objects. The noneffective classes define information associated with various effective classes. By this restriction, only effective classes can be used as base classes. The effective flag is “set” (turned *On*) for only those classes which can be used to provide definition and to create objects. That is, they are active building blocks in which structure rules must be complete. Noneffective classes are used to nonrepetitively define class information that is used by a number of similar classes. Noneffective classes also provide information to be inherited by other object classes. For all noneffective classes this flag is “not set” (turned *Off*). Noneffective classes are used as building blocks only; they are typically used as super classes to define information associated with various effective classes. Noneffective classes are not active and, thus, cannot be used to create objects in the Directory tree.

The base schema contains the following effective classes that can be used to create objects in the Directory tree:

- AFP Server
- Alias
- Bindery Object
- Bindery Queue
- Comm Exec
- Computer
- Country
- Directory Map
- External Entity
- Group
- List
- Locality
- Message Routing Group
- Messaging Server
- NCP Server
- Organization
- Organization Role
- Organization Unit
- Printer
- Print Server
- Profile
- Queue

- Top
- Tree Root
- Unknown
- User
- Volume

The base schema defines the following noneffective classes that cannot be used to create objects in the Directory tree:

- Device
- Organizational Person
- Partition
- Person
- Resource
- Server

Effective or noneffective status is assigned to a class when it is originally defined. That is to say, this value cannot be modified after the class is created. For all effective classes, the naming and containment must be Nonambiguous. For example, if naming attributes have not been specified for a new effective class, naming attributes must be inherited from the new class's super classes. The implication is that naming attribute lists and containment class lists inherited from multiple classes cannot conflict; however, if they are identical, there is no problem. If naming attributes that can be inherited from more than one super class conflict, an effective class must have explicit naming attributes.

#### **Nonremovable Flag**

The *Nonremovable Flag* is “set” (turned *On*) for all base schema object classes. Object classes added to extend the schema are the only ones that may have the nonremovable flag not “set” (turned *Off*).

#### **Ambiguous Naming Flag**

Noneffective classes can be created with ambiguous naming. Only in special cases can effective classes be created with ambiguous naming. For most object classes in the base schema, the *Ambiguous Naming Flag* is turned Off. The only object classes where this flag is turned On are *Top*, *Alias*, *Person*, and *Partition*.

#### **Ambiguous Container Flag**

Noneffective classes can be created with ambiguous containment. Only in special cases can effective classes be created with ambiguous containment. For most object classes in the base schema, the *Ambiguous Container Flag* is turned Off. It is turned On for object classes *Top*, *Alias*, *Person*, and *Partition*.

## Default ACL Templates

Every object has an *Access Control List (ACL)* attribute. This attribute holds information about which trustees have access to the object itself (entry rights), and which trustees have access to the attributes for the object. This information is stored in sets of information containing the trustee name, the affected attribute ([Entry Rights], [All Attributes Rights], or a specific attribute), and the privileges. Some object classes have a default set of values for their *ACL*. The *Default ACL Templates* are included in the individual object class definitions in this section. The inherited default ACL templates are not listed.

## Super Classes

Another type of structuring is done with the *Super Class* designations. This designates the structure of the schema itself, not of the Directory tree. The association of class definitions is established through use of super class designations. Each new object must have a super class list. The complete definition of each object class is derived from the components of the object class itself plus the components of all classes in its super class lineage. The only class that has no super class is *Top*, which is a super class of all other classes. That is, it has no super classes and is used to specify information that pertains to all other classes. The class *Top* is an effective class, but it is a special super class because it cannot be used to define an instance of an object. An object class inherits all the features of its super classes. Hierarchies of classes develop through class inheritance in this manner. The classes at the top of the hierarchy provide general characteristics, while those at the bottom become more and more specialized. The complete set of rules for an object class is called the *Expanded Class Definition*.

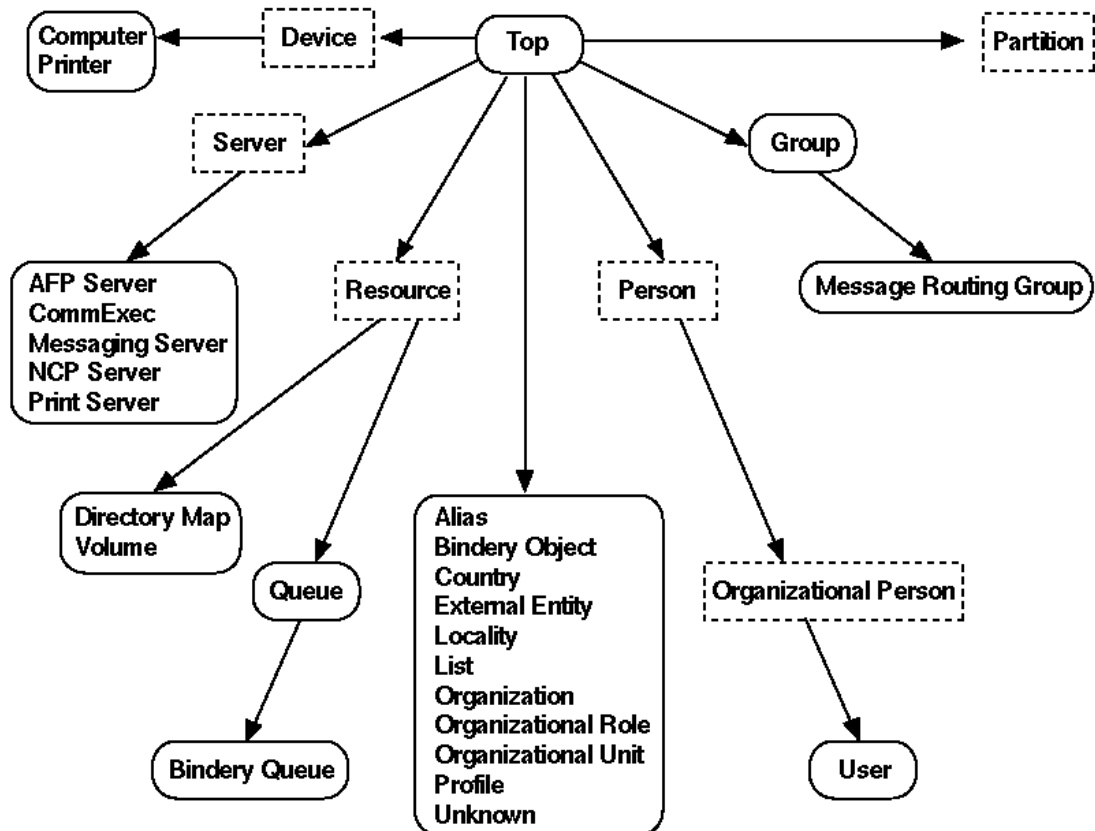
The class from which an object is created is called the object's *Base Class*. The information associated with an object includes the base class and the sum of the information specified by all its super classes. For the purpose of searching the Directory, an object is considered a member of all of its super classes.

You should note that the hierarchy of object classes is distinct from the hierarchy of object names in the Directory tree. Although the schema is stored with the rest of the Directory, schema data is logically separated from the Directory and must be accessed through different functions. Also, the schema's class hierarchy does not form a simple tree graph because a class can be derived from two or more classes that are superior to it.

An object class inherits information types and attributes from its super classes. Thus, super classes control inheritance. That is to say, class inheritance is determined by a class's super class list. Super classes may be multi-valued and are nonrecursive. The condition where a class inherits information from more than one super class is called *Multiple Inheritance*.

Figure 4.3 provides a single graphic view of the base schema, showing the object classes in the structure of the class hierarchy. This provides a visual view of the object classes, super classes, and inheritance. In this view, the arrows show the direction of flow for inheritance. An object class inherits the rules and attributes defined by all its super classes, but does not inherit from its subordinates. Effective object classes are shown with solid line boxes, and noneffective object classes are represented by broken line boxes. The object class *Tree Root* is not shown in this graphical view because it is a virtual object class occupying the same level as *Top*, but inheriting rules and attributes from *Top*.

Figure 4.3  
Directory Tree Class Inheritance



Below is a list of rules that regulate the construction of new object classes. Clients that need to define the new classes should pay close attention to these rules.

1. Object class definitions cannot be recursive. That is to say, an object cannot have itself as a super class.
2. Only classes with complete structure rules can be flagged as effective, and thus used to create objects. This means the super classes, containment, and naming attributes must be complete.

3. An effective class can be constructed in three ways:
  - The class defines its own structure rules
  - The class inherits structure rules from its super classes
  - The class defines part of the structure rules (such as naming) and inherits the other part of the structure rules (such as containment) from a super class
4. Structure rules that might be inherited from its super classes are ignored for a class that defines its own structure rules.
5. If structure rules of an effective class are inherited, they must be nonambiguous.

### **Object Class Attributes**

The attributes assigned to an object class can be mandatory or optional. A client cannot associate an attribute with an object if the attribute is not a mandatory or optional attributes in the object's expanded base class definition. If a client must associate an attribute with a particular object and the attribute is not specified by the object class, the client must either add the attribute to the class as an optional attribute or define a new class that inherits the attribute from the original class.

#### **Mandatory Attributes (Must Contain)**

The attributes listed here are required. If an attribute is mandatory, every object of the class must be assigned a value of that attribute type when it is created. Mandatory attributes can be inherited from the super classes.

#### **Optional Attributes (May Contain)**

The attributes listed here are not required. If an attribute is optional, the client creating the object can assign the attribute a value, but is not required to do so unless that attribute is the only attribute used for naming the object. Optional attributes can also be inherited from the super classes.

## **Reading the Object Class Definitions**

This section defines the object classes that are defined in the nbinit.c file and appear in the Directory database for NDS release 611. For each object class definition, the name of the class appears in large type at the top of the page, followed by a brief description of how the class is used. Each object class then has the following information defined:

### **Class Flags**

The five class flags are presented in a table that indicates the *On* or *Off* status for each flag. Keep in mind that *On* means the flag is “set”, and *Off* means the flag is “not set.”

**Default ACL Template**

This section of the object class definition lists only those default ACLs the object defines. This template is presented in table format, giving the object name, the default rights, and the affected attributes.

**Subclass of**

The immediate super class of the object class is given here, even though an object class may have more than one super class that attributes are inherited from. Those other super classes with the attributes inherited from them are presented in the attributes section.

**Containment**

Objects of the current class can be created as subordinates in the Directory tree to objects of the classes listed here. An object of this class cannot be subordinate to any object that is of a class not listed here.

**Named By**

Listed here are the attributes, mandatory or optional, from which the partial name or *Relative Distinguished Name* (RDN) of objects is derived. At least one must be given a value when creating an object of the class. If the only *Named By* attribute is optional, it is in effect mandatory.

**Attributes**

The attributes are presented in table format with mandatory attributes, those that the object class *Must Contain*, listed first. This is followed by optional attributes, those that the class *May Contain*. The attributes that are unique to the object class are listed in the first column. Those attributes that are inherited from super classes are listed in the other columns to the right with the column headings indicating the super class from which the attributes are inherited.

**Remarks**

This section may include such information as further restrictions of the object class, how to use the class, notes on attributes, and references to related documents.

**Individual Object Class Definitions**

This section contains the detailed definition for each of the 33 object classes used by NDS release 611 objects.

## AFP Server

**Description** This class identifies objects that provide AFP services.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Server

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Inherited From Super Class</b>	
	<b>Unique To Class</b>	<b>Top Server</b>
Must Contain	None	Object Class CN (Common Name)
May Contain	Serial Number Supported Connections	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision Account Balance Allow Unlimited Credit Description Full Name Host Device L (Locality Name) Minimum Account Balance Network Address O (Organization Name) OU (Organizational Unit Name) Private Key Public Key Resource Security Equals Security Flags See Also Status User Version

**Remarks** None



## Alias

**Description** This class defines Alias objects. The aliased object class determines how the Alias is named and where it can be contained.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	OFF	ON	ON	ON	ON

**Default ACL Template** None

**Subclass of** Top

**Containment** Special

**Named By** Special

Attributes	Unique To Class	Inherited From Super Class
Must Contain	Aliased Object Name	Object Class
May Contain	None	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** An *Alias* is a name containing at least one RDN that is an Alias entry. Aliases permit Directory entries to have multiple immediate superiors and consequently provide a basis for alternative names. An Alias typically expresses an alternative relationship to a different hierarchy of objects.

An object in the Directory tree may have zero or more aliases. Several Alias entries may point to the same object entry. Only object entries can have aliases, thus, aliases of aliases are not permitted. An object entry does not have to be a leaf entry to have an Alias. However, Alias entries cannot have subordinates and are always leaf entries.

The Directory uses the *Aliased Object Name* attribute in an Alias entry to identify and to find the corresponding object entry. The *Alias* object class does not specify naming attributes for Alias entries, nor does the class define where Alias entries can be contained in the Directory tree. The Directory enforces the naming and containment rules mandated by the base class of the object the Alias points to.

## Bindery Object

**Description** This class represents an object that Bindery Services has created to emulate bindery objects.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top

**Containment** Organization  
Organizational Unit

**Named By** Bindery Type  
CN (Common Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>
Must Contain	Bindery Object Restriction Bindery Type CN (Common Name)	Object Class
May Contain	Special Case	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The special case notation for optional attributes indicates that any attribute type not listed under “Mandatory Attributes” above is optional for bindery objects.

## Bindery Queue

**Description** This class represents an object that Bindery Services has created to emulate user-defined queue objects.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Queue

**Containment** None

**Named By** Bindery Type  
CN (Common Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>		
		<b>Top</b>	<b>Resource</b>	<b>Queue</b>
Must Contain	Bindery Type	Object Class	CN (Common name)	Queue Directory
May Contain	None	ACL	Description	Device
		Authority Revocation	Host Resource	Host Server
		Back Link	Name	Network Address
		Bindery Property	L (Locality Name)	Operator
		CA Private Key	O (Organization	Server
		CA Public Key	Name)	User
		Certificate Revocation	OU (Organizational	Volume
		Interval	Unit Name)	
		Certificate Validity	See Also	
		Cross Certificate Pair		
		Equivalent To Me		
		Last Referenced		
		Time		
		Obituary		
		Reference		
		Revision		

**Remarks** The Queue Management Service (QMS) requires the host server to identify the NCP server.

The *Operator*, *Server*, and *User* attributes provide the Access Control Lists. The *Operator* attribute identifies users or groups that have operator privileges. The *Server* attribute identifies print servers allowed to service this queue. The *User* attribute contains a list of objects that are authorized to use this queue. The server that controls the queue must determine if the administrator maintains the user list or if the server automatically generates the user list. If the server uses the user list as an Access Control List, the administrator usually maintains the list. If the user list is purely informational, reflecting access control information stored elsewhere, the server usually maintains the list.

The *Queue Directory* attribute names the subdirectory where this queue's files are stored. The *Host Resource Name* attribute is used when the host's local identification for a resource differs from the global resource identification.

The *L (Locality Name)*, *O (Organization Name)* and *OU (Organizational Unit Name)* attributes are useful when multiple localities, organizations, or organizational units use a single resource. If these attributes contain appropriate values, a search can be initiated for resources associated with a particular locality name or organization.

The *Network Address* attribute (inherited from the *Resource* super class) acts as a cache for the server's network address. The user can contact the server without having to dereference the *Host Server* attribute. The *See Also* attribute might be used to list related queues.

## CommExec

**Description** NetWare MHS services use this class.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Server

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Inherited From Super Class</b>	
	<b>Unique To Class</b>	<b>Top Server</b>
Must Contain	None	Object Class CN (Common Name)
May Contain	Network Address Restriction	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision Account Balance Allow Unlimited Credit Description Full Name Host Device L (Locality Name) Minimum Account Balance Network Address O (Organization Name) OU (Organizational Unit Name) Private Key Public Key Resource Security Equals Security Flags See Also Status User Version

**Remarks** None

## Computer

**Description** This class represents computers used as NetWare servers and as client workstations.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Device

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>	
		<b>Top</b>	<b>Device</b>
Must Contain	None	Object Class	CN (Common Name)
May Contain	Operator Server Status	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision	Description L (Locality Name) Network Address O (Organization Name) OU (Organizational Unit Name) Owner See Also Serial Number

**Remarks** The *Operator* attribute lists individuals or groups that are responsible for day-to-day maintenance of the computer. This can differ from the value of *Owner* (inherited from the *Device* super class), which can indicate more of an administrative responsibility with respect to the computer.

The *Server* attribute provides a list of servers that are hosted on this computer. The *See Also* attribute might be used, in this instance, to identify other related computers assigned to a network.

The *L (Locality Name)* attribute identifies the physical location of a device. Although the *O (Organization Name)* and *OU (Organizational Unit Name)* attributes may be present in the device's Distinguished Name, they are repeated to aid searching when an organization spans multiple subtrees in the Directory tree. However, the Directory does not add these attributes automatically, even though they may be present in the device's Distinguished Name. Additional values for the organization

name or organizational unit name may be useful when a device is “co-owned” by multiple organizations.

## Country

**Description** This class defines country entries in the Directory tree.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	ON	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Top

**Named By** C (Country Name)

Attributes	Unique To Class	Inherited From Super Class
Must Contain	C (Country Name)	Object Class
May Contain	Description	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** Countries usually appear as immediate subordinates of the [root] object, of which the base class is *Top*.

The *Description* attribute might contain the full name of the country, because the *Country Name* attribute is restricted to the two-letter code defined by ISO 3166.



## Device

**Description** This class represents physical units that can communicate, such as a modem, printer, etc.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	OFF	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	CN (Common Name)	Object Class
May Contain	Description L (Locality Name) Network Address O (Organization Name) OU (Organizational Unit Name) Owner See Also Serial Number	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** At least one of the attributes *L (Locality Name)*, *Serial Number*, or *Owner* should be included with the object entry, depending on the type of device. The *L (Locality Name)* identifies the physical location of a device.

The *O (Organization Name)* and *OU (Organizational Unit Name)* attributes may already be present in the device's Distinguished Name. However, they are repeated here to aid searching when an organization spans multiple subtrees in the Directory tree. These attributes are not added automatically by the Directory tree, even though they may be present in the device's *Distinguished Name*. Additional values for the *O (Organization Name)* or *OU (Organizational Unit Name)* may be useful when a device is "co-owned" by multiple organizations.

## Directory Map

**Description** This class represents the physical name of the file system directory path.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Resource

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Inherited From Super Class</b>	
	<b>Unique To Class</b>	<b>Resource</b>
Must Contain	Host Server	Object Class CN (Common Name)
May Contain	Path	ACL Description Authority Revocation Host Resource Name Back Link L (Locality Name) Bindery Property O (Organization Name) CA Private Key OU (Organizational Unit Name) CA Public Key See Also Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The *Host Server* attribute identifies the server that owns and services the resource. Requests to manipulate a particular resource must usually be directed to the host server.

The *Host Resource Name* attribute is used when the host's local identification for a resource differs from the global resource identification.

The *L (Locality)*, *O (Organization Name)* and *OU (Organizational Unit Name)* attributes are useful when a resource is used by multiple localities, organizations and organizational units. If these attributes contain appropriate values, a search can be initiated for resources associated with a particular name or organization.

## External Entity

**Description** This class defines non-native NDS objects in the Directory tree.

**Class Flags**

Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
OFF	ON	ON	OFF	OFF

**Default ACL Template**

Object Name	Default Rights	Affected Attributes
[Public]	Read	External Name

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** (CN) Common Name  
OU (Organizational Unit Name)

**Attributes**

	Unique To Class	Inherited From Super Class Top
Must Contain	CN (Common Name)	Object Class
May Contain	Description	ACL
	EMail Address	Authority Revocation
	External Name	Back Link
	Fascimile Telephone Number	Bindery Property
	L (Locality Name)	CA Private Key
	Mailbox ID	CA Public Key
	Mailbox Location	Certificate Revocation
	OU (Organizational Unit Name)	Certificate Validity Interval
	Physical Delivery Office Name	Cross Certificate Pair
	Postal Address	Equivalent To Me
	Postal Code	Last Referenced Time
	Postal Office Box	Obituary
	S (Street or Province Name)	Reference
	SA (Street Address)	Revision
	See Also	
	Title	

**Remarks**

The *External Entity* object can be used by services that need to store information about entities outside of the Directory. A messaging service can use *External Entity* objects to store information about E-Mail users that exist on other systems. In this case the *External Entity* class can be used to store address information about those E-Mail users.

## Group

**Description** This class defines values representing an unordered set of names. The names themselves can represent individual objects or other groups of names.

**Class Flags**

Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
OFF	ON	ON	OFF	OFF

**Default ACL Template**

Object Name	Default Rights	Affected Attributes
[Root]	Read	Member

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

**Attributes**

	Unique To Class	Inherited From Super Class Top
Must Contain	CN (Common Name)	Object Class
May Contain	Description	ACL
	E-Mail Address	Authority Revocation
	Full Name	Back Link
	GID (Group ID)	Bindery Property
	L (Locality Name)	CA Private Key
	Login Script	CA Public Key
	Mailbox ID	Certificate Revocation
	Mailbox Location	Certificate Validity Interval
	Member	Cross Certificate Pair
	O (Organization Name)	Equivalent To Me
	OU (Organizational Unit Name)	Last Referenced Time
	Owner	Obituary
	Profile	Reference
	Profile Membership	Revision
	See Also	

**Remarks** A *Group's* membership is static; that is, it is explicitly modified by administrative action, rather than dynamically determined each time the group is referred to. The membership of a group can be reduced to a set of individual objects' names by replacing each group with its membership. This process could be carried out recursively until all constituent group names have been eliminated and only the names of individual objects remain.

NetWare Directory Service operations in general do not perform recursive membership expansion. However, access control resolution effectively expands one level of group listed in an ACL. Thus, if 'A' is a

member of group 'B,' which is in turn listed in an access control list, 'A' will gain the access granted to group 'B.' However, if 'A' is a member of group 'B,' which is a member of group 'C,' and 'C' is listed in an access control list, 'A' will not gain the access granted to group 'C.' Other applications are free to perform recursive group expansion, if they so choose. Security equivalence is implied for most classes, but it is not actually required to be present for most attributes.

The attributes *L* (*Locality Name*), *O* (*Organization Name*), and *OU* (*Organizational Unit Name*) may already be present in the group's Distinguished Name. They are repeated here to aid searching when an organization spans multiple subtrees in the Directory tree. Additional values for the *L*, *O* and *OU* attributes can be useful when a group contains members from multiple organizations, organizational units and localities.

The *Owner* attribute could be used to contain the name of the group leader or group moderator. This value might not be the same as the set of individuals authorized to modify the group object.

The *See Also* attribute might be used to list related groups.

## List

**Description** This class represents an unordered set of names in the Directory.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

<b>Default ACL Template</b>	<b>Object Name</b>	<b>Default Rights</b>	<b>Affected Attributes</b>
	[Public]	Read	Member

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	CN (Common Name)	Object Class
May Contain	Description	ACL
	E-Mail Address	Authority Revocation
	Full Name	Back Link
	L (Locality Name)	Bindery Property
	Mailbox ID	CA Private Key
	Mailbox Location	CA Public Key
	Member	Certificate Revocation
	O (Organization Name)	Certificate Validity Interval
	OU (Organizational Unit Name)	Cross Certificate Pair
	Owner	Equivalent To Me
	See Also	Last Referenced Time
		Obituary
		Reference
		Revision

**Remarks** The *Member* attribute contains the object names that are members of the list. The members can be individual objects, including *Group* objects, or the names of other *List* objects. Unlike group membership, list membership does not imply security equivalence.

## Locality

**Description** This class defines geographic locations in the Directory tree.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	ON	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Country  
Locality  
Organization  
Organizational Unit

**Named By** L (Locality Name)  
S (State or Province Name)

Attributes	Unique To Class	Inherited From Super Class Top
Must Contain	None	Object Class
May Contain	Description L (Locality Name) S (State or Province Name) SA (Street Address) See Also	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** At least one of the attributes, *L (Locality Name)* or *S (State or Province Name)* must be represented since these are included in the naming rules. We recommend that only an object contained by *Country* use the *S (State or Province Name)* attribute as a naming attribute.

## Message Routing Group

**Description** This class represents a group or cluster of messaging servers that communicate directly with each other for transferring messages.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

<b>Default ACL Template</b>	<b>Object Name</b>	<b>Default Rights</b>	<b>Affected Attributes</b>
	[Self]	Browse	[Entry Rights]
	[Self]	Read	All Attributes

**Subclass of** Group

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Inherited From Super Class</b>	
	<b>Unique To Class</b>	<b>Group</b>
Must Contain	None	Object Class CN (Common Name)
May Contain	None	ACL Description Authority Revocation EMail Address Back Link Full Name Bindery Property GID CA Private Key L (Locality Name) CA Public Key Login Script Certificate Revocation Mailbox ID Certificate Validity Interval Mailbox Location Cross Certificate Pair Member Equivalent To Me O (Organization Name) Last Referenced Time OU (Organizational Unit Name) Obituary Owner Reference Profile Revision Profile Membership See Also

**Remarks** The *Member* attribute (inherited from *Group*) lists the messaging servers that belong to the *Message Routing Group*.

The *Owner* attribute (inherited from *Group*) contains the name of the postmaster general of the messaging server's message routing group. The owner has the authority to add a messaging server's name to, or remove a messaging server's name from, the member list.



## Messaging Server

**Description** This class represents messaging servers, such as NetWare MHS servers, in the Directory tree.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	OFF	ON	ON	OFF	OFF

Default ACL Template	Object Name	Default Rights	Affected Attributes
	[Self]	Browse	[Entry Rights]
	[Self]	Read	All Attributes
	[Public]	Read	Messaging Server Type
	[Public]	Read	Messaging Database Location

**Subclass of** Server

**Containment** None

**Named By** None

Attributes	Inherited From Super Class	
	Unique To Class	Top Server
Must Contain	None	Object Class CN (Common Name)
May Contain	Message Routing Group Messaging Database Location Messaging Server Type Postmaster Supported Gateway Supported Services	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision Account Balance Allow Unlimited Credit Description Full Name Host Device L (Locality Name) Minimum Account Balance Network Address O (Organization Name) OU (Organizational Unit Name) Private Key Public Key Resource Security Equals Security Flags See Also Status User Version

**Remarks** A MHS messaging server picks up messages, which are either submitted by messaging applications (such as E-Mail) or transferred from another messaging server, and delivers them to the recipients. For recipients whose mailboxes are local on the messaging server, the messages are delivered to their mailboxes. Otherwise, the messaging server transfers

the message to another messaging server for eventual delivery to the recipient's mailbox.

A MHS messaging server runs as an NLM (MHS.NLM) on a NetWare server. It is represented by an NDS leaf object whose object class is *Messaging Server*. There is no limit to the number of mailboxes it serves, as long as there is enough disk space for the mailboxes.

The *Message Routing Group* attribute names the message routing groups to which the messaging server belongs. The *Messaging Database Location* names the volume and path on which the message directory (such as SYS:MHS) resides. MHS messaging servers use a file system subtree to:

- Receive messages from applications, other messaging servers, and gateways.
- Store messages while they are being routed.
- Store internal control files.
- Extract files.

The *Messaging Server Type* attribute identifies the type of the messaging server object, such as MHS, GMHS, or X400. The *Postmaster* specifies one or more users who have the privileges to manage the messaging server, such as privileges to remove a mailbox. Postmasters also receive messages about special events in the messaging server, such as messages being processed.

The *Supported Gateway* attribute specifies the messaging gateways that are connected through the messaging server. It provides a messaging connectivity between the MHS messaging system and foreign messaging systems. The *Supported Services* attribute indicates the messaging capabilities of the server.

The *Host Device* attribute (inherited from *Server*) identifies the NCP server on which the messaging server's software runs. The *User* (also inherited from *Server*) contains a list of users whose mailboxes are serviced by the messaging server. Any effective object that has the *Mailbox ID* and *Mailbox Location* attributes is a valid value for this list. Adding an object to the *User* list has the same effect as assigning values to the object's *Mailbox ID* and *Mailbox Location* attributes. An administrator can give an object a mailbox by either means.

## NCP Server

**Description** This class represents servers that provide NCP transport and sessions services.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Server

**Containment** None

**Named By** None

Attributes	Inherited From Super Class	
	Unique To Class	Top Server
Must Contain	None	Object Class CN (Common Name)
May Contain	DS Revision Messaging Server Operator Supported Services	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision Account Balance Allow Unlimited Credit Description Full Name Host Device L (Locality Name) Minimum Account Balance Network Address O (Organization Name) OU (Organizational Unit Name) Private Key Public Key Resource Security Equals Security Flags See Also Status User Version

**Remarks** Note that the individual services on an NCP server do not need distinct Directory names because they can all share a common NCP session. However, individual resources on a server can require distinct Directory entries.

The *Supported Services* attribute lists NCP-based services or features available at this network address. It should not be used to list other non-NCP services residing on the same host.

The *Operator* attribute is used by the NCP server as an access control list. If an object is listed in this attribute, that object is allowed to perform remote console operations.

The *NCP Server* class is intended to represent both Bindery-based and NDS-based NCP servers. The *Version* attribute (inherited from the *Server* super class) should distinguish one type of server from the other.

The *Private Key* and *Public Key* attributes are present if the server is a client of the Directory's Authentication Services. The *Resource* attribute contains a list of resources managed by this server.

The *User* attribute contains a list of objects that are authorized to use this server. The server must determine if an administrator is to maintain the user list, or if the server is to automatically generate the list. If the server uses the user list as an ACL, the administrator usually maintains the list. If the user list is purely informational, reflecting access control information stored elsewhere, the server usually maintains the list.

## Organization

**Description** This class defines organization objects in the Directory tree.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	ON	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Country  
Locality  
Top, Tree Root

**Named By** O (Organization Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	O (Organization Name)	Object Class
May Contain	Description Detect Intruder EMail Address Facsimile Telephone Number Intruder Attempt Reset Interval Intruder Lockout Reset Interval L (Locality Name) Lockout After Detection Login Intruder Limit Login Script Mailbox ID Mailbox Location NSS Domain Physical Delivery Office Name Postal Address Postal Code Postal Office Box Print Job Configuration Printer Control S (State or Province Name) SA (Street Address) See Also Telephone Number	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** An organization located directly under the [Root] object denotes an international organization. For international organizations, the values of the *O (Organization Name)* attribute must all be unique.

## Organizational Person

**Description** This class defines objects representing people employed by, or in some other important ways associated with, an organization.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	OFF	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Person

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)  
OU (Organizational Unit Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>	
		<b>Top</b>	<b>Person</b>
Must Contain	None	Object Class	CN (Common Name) Surname
May Contain	EEmail Address Facsimile Telephone Number L (Locality Name) Mailbox ID Mailbox Location OU (Organizational Unit Name) Physical Delivery Office Name Postal Address Postal Code Postal Office Box S (State or Province Name) SA (Street Address) Title	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision	Description Full Name Generational Qualifier Given Name Initials See Also Telephone Number

**Remarks** The X.500 standard defines two subclasses of *Person*: *Organizational Person* and *Residential Person*. The schema defined by this document does not include *Residential Person*, but the division of *Person* from *Organizational Person* has been maintained for future compatibility with X.500.

## Organizational Role

**Description** This class defines a position or role within an organization.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	CN (Common Name)	Object Class
May Contain	Description EMail Address Facsimile Telephone Number L (Locality Name) Mailbox ID Mailbox Location OU (Organizational Unit Name) Physical Delivery Office Name Postal Address Postal Code Postal Office Box Role Occupant S (State or Province Name) SA (Street Address) See Also Telephone Number	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** Normally, an organizational role is thought to be performed by a particular organizational person. Over its lifetime, however, an organizational role may be filled by a succession of different organizational people. In general, an organizational role may be filled by a person or a nonhuman entity.

## Organizational Unit

**Description** This class defines subdivisions of organizations.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	ON	ON	ON	OFF	OFF

<b>Default ACL Template</b>	<b>Object Name</b>	<b>Default Rights</b>	<b>Affected Attributes</b>
	[Self]	Read	Login Script

**Subclass of** Top, Tree Root

**Containment** Locality  
 Organization  
 Organizational Unit

**Named By** OU (Organizational Unit Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	OU (Organization Unit Name)	Object Class
May Contain	Description	ACL
	Detect Intruder	Authority Revocation
	EMail Address	Back Link
	Facsimile Telephone Number	Bindery Property
	Intruder Attempt Reset Interval	CA Private Key
	Intruder Lockout Reset Interval	CA Public Key
	L (Locality Name)	Certificate Revocation
	Lockout After Detection	Certificate Validity Interval
	Login Intruder Limit	Cross Certificate Pair
	Login Script	Equivalent To Me
	Mailbox ID	Last Referenced Time
	Mailbox Location	Obituary
	NSS Domain	Reference
	Physical Delivery Office Name	Revision
	Postal Address	
	Postal Code	
	Postal Office Box	
	Print Job Configuration	
	Printer Control	
	S (State or Province Name)	
	SA (Street Address)	
	See Also	
	Telephone Number	

**Remarks** None



## Partition

**Description** This class encapsulates the information required to maintain the synchronization and connectivity of the Directory's distributed operations.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	OFF	ON	ON	ON

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>
Must Contain	None	Object Class
May Contain	Authority Revocation CA Private Key CA Public Key Certificate Revocation Convergence Cross Certificate Pair High Convergence Sync Interval Inherited ACL Low Convergence Reset Time Low Convergence Sync Interval Partition Control Partition Creation Time Received Up To Replica Replica Up To Synchronized Up To	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The name server automatically adds this class and its required attributes to a partition's root object. This class is added in addition to the base class of the object, but does not change that base class. Most of the partition attributes are operational in nature, so the name server can supply initial values for these attributes automatically.

The optional attributes *Authority Revocation*, *CA Private Key*, *CA Public Key*, *Certificate Revocation*, *Convergence*, and *Cross Certificate Pair* were originally only in the *Partition* class; however, for NDS release 489 they were also added to the *Top* object class. Thus, they are now redundant in the *Partition* class because they are inherited from *Top*.

## Person

**Description** This class represents the common elements of *Organizational Person* and *Residential Person* objects.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	OFF	ON	ON	ON

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>
Must Contain	CN (Common Name) Surname	Object Class
May Contain	Description Full Name Generational Qualifier Given name Initials See Also Telephone Number	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The X.500 standard defines two subclasses of *Person*: *Organizational Person* and *Residential Person*. The schema defined by this document does not include *Residential Person*, but the division of *Person* from *Organizational Person* has been maintained for future compatibility with X.500.

## Print Server

**Description** This class represents NetWare print servers.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Server

**Containment** None

**Named By** None

Attributes	Inherited From Super Class	
	Unique To Class	Top Server
Must Contain	None	Object Class CN (Common Name)
May Contain	Operator Printer SAP Name	ACL Account Balance Allow Unlimited Credit Description Full Name Host Device L (Locality Name) Minimum Account Balance Network Address O (Organization Name) OU (Organizational Unit Name) Private Key Public Key Resource Security Equals Security Flags See Also Status User Version

**Remarks** The *Print Server* class differs from the *NCP Server* class in that print servers use the SPX protocol for communication, rather than NCP. The *Print Server* class is used for both Bindery-based and NDS-based print servers.

The *Private Key* and *Public Key* attributes are present only if the server is a client of the Directory's Authentication Service. Bindery-based print servers will not have the *Public Key* and *Private Key* attributes.

The *Version* attribute (inherited from *Server* super class) indicates whether the server is Bindery-based or NDS-based. The *Operator* and *User* attributes (inherited from *Server* super class) are used by the print server as access control lists. The *Operator* attribute identifies those

individuals who are authorized to act as print server operators. The *Queue* attribute lists the queues that are serviced by this print server.

The *Host Device* attribute identifies the device that hosts the server. This is usually a computer, but may be some other device such as a printer hosting a build-in print server. The *Resource* attribute contains a list of resources managed by the service.

The *User* attribute contains a list of user objects (individuals) authorized to use the print server. The server must determine if an administrator is to maintain the user list, or if the server is to automatically generate the list. If the server uses the user list as an access control list, the administrator usually maintains the list. If the user list is purely informational, reflecting access control information stored elsewhere, the server usually maintains the list.

## Printer

**Description** This class represents printers in the Directory tree. A *Printer* object points to the queues it is attached to.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Device

**Containment** None

**Named By** None

Attributes	Inherited From Super Class	
	Unique To Class	Top Device
Must Contain	None	Object Class CN (Common Name)
May Contain	Cartridge Default Queue Host Device Memory Network Address Restriction Notify Operator Page Description Language Print Server Printer Configuration Queue Status Supported Typefaces	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The *O* (*Organization Name*) and the *OU* (*Organizational Unit Name*) attributes may already be present in the device's Distinguished Name. They are repeated here to aid searching when an organization spans multiple subtrees in the Directory tree. However, the Directory does not add these attributes automatically, even though they may be present in the device's Distinguished Name. Additional values for the *O* or *OU* attributes may be useful when multiple organizations "co-own" a device.

The *Host Device* attribute in this class denotes the computer (or other device) to which the printer is attached. The *Host Server* attribute identifies the print servers that manage this device.

The *Supported Typefaces* and *Page Description Language* attributes are included to aid in a search for printers with a particular set of capabilities. The contents of these attributes are statically maintained by an

administrator, rather than being dynamically updated from printer feedback.

The *Queue* attribute identifies the associated queues through which the printer may be accessed. The *L (Locality Name)* attribute can be used to identify a device's physical location.

## Profile

**Description** This class specifies a shared login configuration.

Class Flags	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
	OFF	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

Attributes	Unique To Class	Inherited From Super Class Top
Must Contain	CN (Common Name) Login Script	Object Class
May Contain	Description Full Name L (Locality Name) O (Organization Name) OU (Organizational Unit Name) See Also	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** A *Profile* class has an associated *Login Script* attribute that contains the standard login script. This login script is executed after the container login script, but before the user login script. The *Profile* should contain group-related conditional statements.

## Queue

**Description** This class represents batch-processing queues available in the NetWare NCP environment.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

<b>Default ACL Template</b>	<b>Object Name</b>	<b>Default Rights</b>	<b>Affected Attributes</b>
	[Root]	Read	All Attributes

**Subclass of** Resource

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Inherited From Super Class</b>	
	<b>Unique To Class</b>	<b>Resource</b>
Must Contain	Queue Directory	Object Class CN (Common Name)
May Contain	Device	ACL Description
	Host Server	Authority Revocation Host Resource Name
	Network Address	Back Link Locality Name
	Operator	Bindery Property O (Organization Name)
	Server	CA Private Key OU (Organizational Unit Name)
	User	CA Public Key See Also
	Volume	Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The *Host Server* attribute (inherited from *Resource* super class) identifies the NCP server that provides the Queue Management Service (QMS) for this queue. Requests to manipulate a particular resource must usually be directed to the host server. The *Queue Directory* attribute names the subdirectory where this queue's files are stored.

The *Host Resource Name* attribute is used when the host's local identification for a resource differs from the global resource identification. The *Network Address* attribute acts as a cache for the server's network address. The user can contact the server without having to dereference the *Host Server* attribute.

The *Operator*, *Server* and *User* (inherited from *Resource*) attributes are used by the QMS as Access Control Lists. The *Operator* attribute



identifies the users or groups that have operator privileges. The *Server* attribute identifies print servers allowed to service this queue. The *User* attribute contains a list of objects (individuals and groups) that are authorized to use this resource. The server that controls the resource must determine if the user list is maintained by an administrator or if the list is automatically generated by the server. If the server uses the user list as an Access Control List, the administrator usually maintains the list. If the user list is purely informational, reflecting access control information stored elsewhere, the server usually maintains the list.

The *L (Locality Name)*, *O (Organization Name)* and *OU (Organizational Unit Name)* attributes are useful when multiple localities, organizations, or organizational units use a resource. If these attributes contain appropriate values, a search can be initiated for resources associates with a particular locality name or organization.

## Resource

**Description** This class identifies logical resources available on the network.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	OFF	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	CN (Common Name)	Object Class
May Contain	Description Host Resource Name L (Locality Name) O (Organization Name) OU (Organizational Unit Name) See Also	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** The *Resource* class differs from the *Device* class in that a device is a physical unit, such as a printer, and a resource is a nonphysical, logical unit, network resource, such as a printer queue.

The *Host Resource Name* attribute is used when the host's local identification for a resource differs from the global resource identification. The *L (Locality Name)*, *O (Organization Name)*, and *OU (Organizational Unit Name)* attributes are useful when a resource is used by multiple localities, organizations, or organizational units. If these attributes contain appropriate values, a search can be initiated for resources associated with a particular locality name or organization.

## Server

**Description** This class identifies entities that manage one or more resources and provide access to those resources through a communications protocol.

**Class Flags**

Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container
OFF	OFF	ON	OFF	OFF

**Default ACL Template**

Object Name	Default Rights	Affected Attributes
[Public]	Read	Messaging Server
[Public]	Read	Network Address
[Self]	Supervisor	[Entry Rights]

**Subclass of** Top, Tree Root

**Containment** Organization  
Organizational Unit

**Named By** CN (Common Name)

**Attributes**

	Unique To Class	Inherited From Super Class Top
Must Contain	CN (Common Name)	Object Class
May Contain	Account Balance	ACL
	Allow Unlimited Credit	Authority Revocation
	Description	Back Link
	Full Name	Bindery Property
	Host Device	CA Private Key
	L (Locality Name)	CA Public Key
	Minimum Account Balance	Certificate Revocation
	Network Address	Certificate Validity Interval
	O (Organization Name)	Cross Certificate Pair
	OU (Organizational Unit Name)	Equivalent To Me
	Private Key	Last Referenced Time
	Public Key	Obituary
	Resource	Reference
	Security Equals	Revision
	Security Flags	
	See Also	
	Status	
	User	
	Version	

**Remarks** The *Host Device* attribute identifies the device that hosts the server. This is usually a computer but might be some other device. For example, a printer could host a built-in print server.

The *Private Key* and *Public Key* attributes are present if the server is a client of the Directory's Authentication Services. The *Resource* attribute contains a list of resources managed by this service.

The *User* attribute contains a list of user objects that are authorized to use this server. The server must determine if an administrator is to maintain the user list, or if the server automatically generates the list. If the server uses the user list as an Access Control List, the administrator usually maintains the list. If the user list is purely informational, reflecting access control information stored elsewhere, the server usually maintains the list.

## Top

<b>Description</b>	This class is the super class of all other object classes. It is the class for the Directory tree's [Root] object.										
<b>Class Flags</b>	<table border="1"> <thead> <tr> <th>Container</th> <th>Effective</th> <th>Nonremovable</th> <th>Ambiguous Naming</th> <th>Ambiguous Container</th> </tr> </thead> <tbody> <tr> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> </tr> </tbody> </table>	Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container	ON	ON	ON	ON	ON
Container	Effective	Nonremovable	Ambiguous Naming	Ambiguous Container							
ON	ON	ON	ON	ON							
<b>Default ACL Template</b>	<table border="1"> <thead> <tr> <th>Object Name</th> <th>Default Rights</th> <th>Affected Attributes</th> </tr> </thead> <tbody> <tr> <td>[Creator]</td> <td>Supervisor</td> <td>[Entry Rights]</td> </tr> </tbody> </table>	Object Name	Default Rights	Affected Attributes	[Creator]	Supervisor	[Entry Rights]				
Object Name	Default Rights	Affected Attributes									
[Creator]	Supervisor	[Entry Rights]									
<b>Subclass of</b>	None										
<b>Containment</b>	None										
<b>Named By</b>	None										
<b>Attributes</b>	<hr/> <p><b>Unique To Class</b></p> <hr/> <p>Object Class  ACL  Authority Revocation  Back Link  Bindery Property  CA Private Key  CA Public Key  Certificate Revocation  Certificate Validity Interval  Cross Certificate Pair  Equivalent To Me  Last Referenced Time  Obituary  Reference  Revision</p> <hr/>										
<b>Remarks</b>	<p>This class mandates that all objects will contain an <i>Object Class</i> attribute. Although <i>Top</i> is an effective class, it is a special case in that the user cannot construct objects from this class.</p> <p>The optional attributes <i>Authority Revocation</i>, <i>CA Private Key</i>, <i>CA Public Key</i>, <i>Certificate Revocation</i>, <i>Convergence</i>, and <i>Cross Certificate Pair</i> were originally only in the <i>Partition</i> class; however, for NDS release 489 they were also added to the <i>Top</i> object class. Thus, they are now redundant in the <i>Partition</i> class because they are inherited from <i>Top</i>.</p>										

## Tree Root

**Description** This class defines the Directory tree's [Root] object.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	ON	ON	ON	OFF	OFF

**Default ACL Template** None

**Subclass of** Top

**Containment** None

**Named By** T (Tree Name)

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>
Must Contain	T (Tree Name)	Object Class
May Contain		ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** Tree Root is synonymous with the [root] of the tree, of which the base class is *Top*.

## Unknown

**Description** This class represents any object created by the server to restore an object whose base class is no longer defined by the schema.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	ON	ON	ON	ON	ON

**Default ACL Template** None

**Subclass of** Top, Tree Root

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class Top</b>
Must Contain	None	Object Class
May Contain	None	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision

**Remarks** Any attribute is valid as an optional attribute for this class. The client cannot create objects of class *Unknown*.

## User

**Description** This class represents users of network services.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

<b>Default ACL Template</b>	<b>Object Name</b>	<b>Default Rights</b>	<b>Affected Attributes</b>
	[Public]	Read	Message Server
	[Root]	Browse	[Entry Rights]
	[Root]	Read	Group Membership
	[Root]	Read	Network Address
	[Self]	Read	All Attributes
	[Self]	Read/Write	Login Script
	[Self]	Read/Write	Print Job Configuration

**Subclass of** Organizational Person

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Unique To Class</b>	<b>Inherited From Super Class</b>		
		<b>Top</b>	<b>Person</b>	<b>Organizational Person</b>
Must Contain	None	Object Class	CN (Common Name) Surname	None
May Contain	Account Balance Allow Unlimited Credit Group Membership Higher Privileges Home Directory Language Last Login Time Locked By Intruder Login Allowed Time Map Login Disabled Login Expiration Time Login Grace Limit Login Grace Remaining Login Intruder Address	ACL Authority Revocation Back Link Bindery Property CA Private Key CA Public Key Certificate Revocation Certificate Validity Interval Cross Certificate Pair Equivalent To Me Last Referenced Time Obituary Reference Revision	Description Full Name Generational Qualifier Given Name Initials See Also Telephone Number	EMail Address Facsimile Telephone Number L (Locality Name) Mailbox ID Mailbox Locality OU (Organizational Unit Name) Physical Delivery Office Name Postal Address Postal Code Postal Office Box S (State or Province Name) SA (Street Address) Title



---

Login Intruder Attempts
Login Intruder Reset Time
Login Maximum Simultaneous
Login Script
Login Time
Message Server
Minimum Account Balance
Network Address
Network Address Restriction
Password Allow Change
Password Expiration Interval
Password Expiration Time
Password Minimum Length.
Password Required
Password Unique Required
Password Used
Print Job Configuration
Printer Control
Private Key
Profile
Profile Membership
Public Key
Security Equals
Security Flags
Server Holds
Type Creator Map
UDI (User ID)

---

**Remarks**

Note that the *User* class includes both client and service providers. In this context, *Private Key* stores the object's private key encrypted by the object's password.

The X.500 standard defines two subclasses of *Person*: *Organizational Person* and *Residential Person*. The schema defined by this document does not include *Residential Person*, but the division of *Person* from *Organizational Person* has been maintained for future compatibility with X.500.

## Volume

**Description** This class represents NetWare file system volumes.

<b>Class Flags</b>	<b>Container</b>	<b>Effective</b>	<b>Nonremovable</b>	<b>Ambiguous Naming</b>	<b>Ambiguous Container</b>
	OFF	ON	ON	OFF	OFF

<b>Default ACL Template</b>	<b>Object Name</b>	<b>Default Rights</b>	<b>Affected Attributes</b>
	[Root]	Read	Host Resource Name
	[Root]	Read	Host Server

**Subclass of** Resource

**Containment** None

**Named By** None

<b>Attributes</b>	<b>Inherited From Super Class</b>	
	<b>Unique To Class</b>	<b>Resource</b>
Must Contain	Host Server	CN (Common Name)
May Contain	Status	Description
		Host Resource Name
		L (Locality Name)
		O (Organization Name)
		OU (Organizational Unit Name)
		See Also
		ACL
		Authority Revocation
		Back Link
		Bindery Property
		CA Private Key
		CA Public Key
		Certificate Revocation
		Certificate Validity Interval
		Cross Certificate Pair
		Equivalent To Me
		Last Referenced Time
		Obituary
		Reference
		Revision

**Remarks** The *Volume* subclass exists primarily to allow volume objects to be distinguished from other types of Resource objects.

When present, the *Host Resource Name* attribute (inherited from *Resource*) contains the local volume name that corresponds to the volume name on the server. This attribute is used when the host's local identification of a resource differs from the global resource identification. If the attribute is not present, the local volume name "SYS:" can be assumed. The *Host Server* attribute identifies the server that owns and services the resource. Requests to manipulate a particular resource must usually be directed to the host server.

The *L (Locality Name)*, *O (Organization Name)*, and *OU (Organizational Unit Name)* attributes are useful when multiple localities, organizations,

or organizational units use a single resource. If these attributes contain appropriate values, a search can be initiated for resources associated with a particular locality name or organization.

