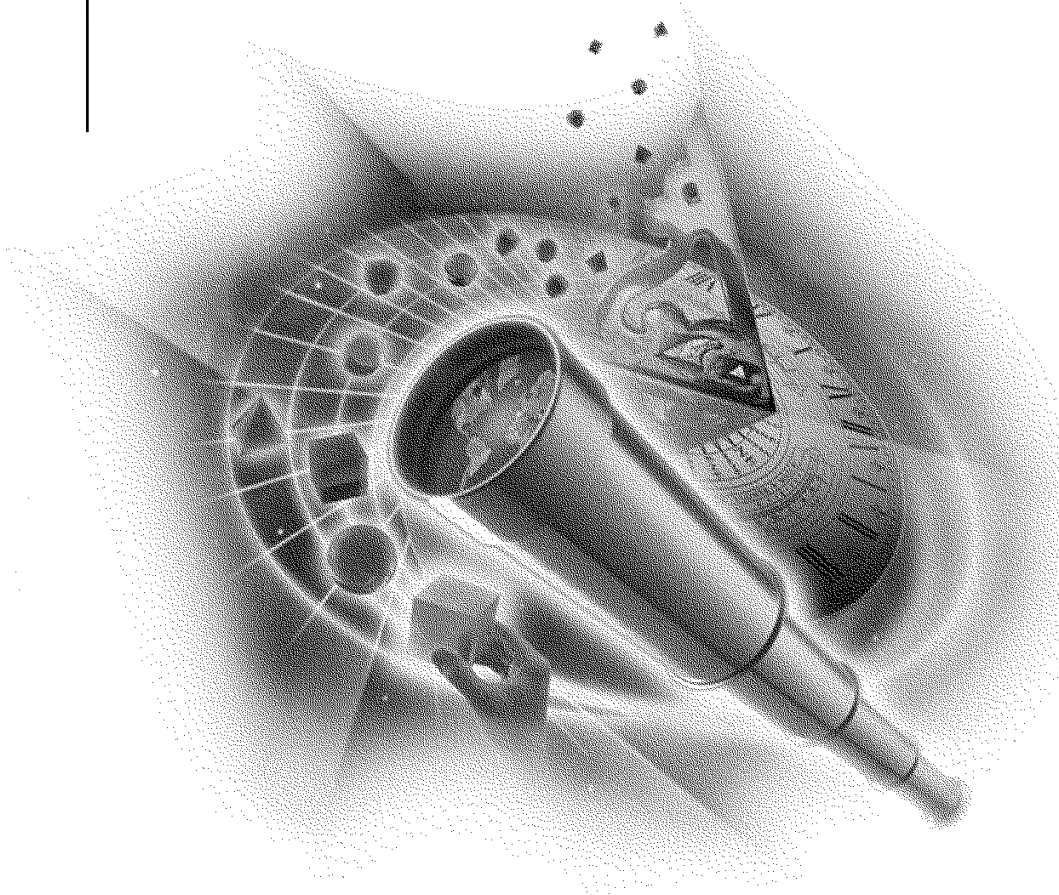


VERSION 8

Managing
the
NDS Schema



NDSTM 8

NEXT GENERATION DIRECTORY SERVICES

Novell[®]

disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

export notice

This product may require export authorization from the U.S. Department of Commerce prior to exporting from the U.S. or Canada.

trademarks

Novell and NetWare are registered trademarks of Novell, Inc. in the United States and other countries. ConsoleOne and NDS are trademarks of Novell, Inc.

Copyright © 1992-1999 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent Nos. 5,870,739; 5,873,079; and 5,884,304. U.S. and International Patents Pending.

**Novell, Inc.
122 East 1700 South
Provo, UT 84606
U.S.A.**

**Managing the NDS Schema
May 1999
104-000100-001**

Contents

Overview

Overview	iii
--------------------	-----

1 Schema Management

Understanding Schema Manager	1
Understanding Schema Classes, Attributes, and Syntaxes	2
Understanding Mandatory and Optional Attributes	4
Planning the Schema	4
Understanding Class Inheritance and Hierarchy	4
Managing the Schema	5

preface

Overview

The Schema Manager tool in ConsoleOne™ allows you to change NDS™ object rules, which gives you the option of creating new object classes and attributes for your organization.

1 **Schema Management**

The schema defines classes of NDS objects (such as Users, Printers and Groups) and which information is required or optional at the time that NDS object is created. Every NDS object has a schema class that has been defined for that type of object.

The base schema is the schema that ships with NDS. The difference between the base schema and the extended schema is simple. Once the base schema has been modified in any way—such as adding a new class or a new attribute—then it is considered the extended schema.

You don't need to extend the schema, but you have the ability to do so. The Schema Manager tool in ConsoleOne allows the schema to be extended as needed to meet organizational needs. For example, you can extend your schema if your organization requires special footwear for employees and you need to keep track of employee shoe sizes. You can create a new attribute called Shoe Size and then add it to the User class.

Understanding Schema Manager

Schema Manager is an integrated snap-in to ConsoleOne. It allows those with supervisor rights to a tree (normally Admin) to customize the schema of that tree. To start Schema Manager from ConsoleOne, make sure you have selected a tree (or an object within a tree) and select Tools > Schema Manager.

Schema Manager has some new features with NDS 8:

- You can create auxiliary classes and associate them with specific objects rather than an entire class. You'll see this option when you use the Create Class wizard.
- ASN1 IDs uniquely identify each attribute and class in the schema.

Warning You should only add a *registered* ASN1 ID to a class or attribute. Otherwise, you should leave this option alone.

Understanding Schema Classes, Attributes, and Syntaxes

A class is like a set of rules for an NDS object. An NDS object is a new record with data built according to the rules of its class.

When you create a new NDS object, you select a class as a starting point in defining that object. The class is like a request form, fitted with the class rules. Once the class is selected, you complete the “request form” in order to provide essential and specific information on the new object.

The class has a class name, an inheritance class (unless it is at the top of the class hierarchy), class flags, and a group of attributes. Classes are named like NDS objects—User, Printer, Queue, Server—yet they are just structure and rules, no content.

The inheritance class is a class that is used as a starting point for defining another class. All of the attributes granted to the inheritance class are inherited by the classes that come below it in the class hierarchy. Inheritance makes defining and modifying new classes simpler than starting a new class from nothing.

Attributes

If a class is like a form, then the attribute is one field on the form. When an attribute is created, it is named (such as “surname” or “employee number”) and given a syntax type (such as “string A-Z, 0-9” or “number -999 to 999”). From then on, it is available in the attribute list.

Syntaxes

There are 29 syntax options from which to choose (available through a drop down box), such as “Case Exact String” and “Integer.” These are used to specify how you want data entered and stored for each attribute. The syntax can only be specified when an attribute is created. You cannot modify it later.

Available syntaxes include

- [Back Link](#)
- [Boolean](#)
- [Case Exact String](#)

- Case Ignore List
- Case Ignore String
- Class Name
- Counter
- Distinguished Name
- Email Address
- Facsimile Telephone Number
- Hold
- Integer
- Interval
- Net Address
- Numeric String
- Object ACL
- Octet List
- Octet String
- Path
- Postal Address
- Printable String
- Replica Pointer
- Stream
- Telephone Number
- Time

- Timestamp
- Typed Name
- Unknown

Understanding Mandatory and Optional Attributes

A class is a group of attributes organized in a meaningful way. Some of these attributes are mandatory and some are optional.

Mandatory Attributes

A mandatory attribute is one that must be completed when an object is being created. For instance, if a new User object is being created using the User class, which has the employee number as a mandatory attribute, then the new User object cannot be created without providing the employee number.

Optional Attributes

An optional attribute is one that can be completed if desired but can be left without content. For instance, if a new User object is being created using the User class, which has Other names as an optional attribute, then the new User object can be created with or without data provided for that attribute-- depending on whether the new user is known by other names.

An exception to the rule is that when an optional attribute is used for naming, the attribute then becomes mandatory.

Planning the Schema

Designing your schema can save you time and effort in the long run. You can view the base schema and determine if it will meet your needs or if modifications are required.

Understanding Class Inheritance and Hierarchy

A class hierarchy shows how a class is associated with a parent class. This is a way of associating similar classes and allowing attributes to be inherited. It also defines the types of containers in which the class is valid.

When creating a new class, you can specify an inheritance class (which will allow the new class to inherit all of the attributes and flags of a class higher in the hierarchy) and then customize the new class by selecting one or more attributes to add to those which were inherited. The additional attributes can be selected as mandatory, naming, or optional attributes.

You can also modify existing classes by adding optional attributes.

Managing the Schema

You can use the Schema Manager tool in ConsoleOne to do the following:

- View information on a class or attribute: Double-click the class or attribute.
- Create a class or attribute: Click Create from the Classes or Attributes page.
- Add an attribute to an existing class: Select a class > Add.
- Delete classes or attributes that you have previously added (if they are not used): Select a class or attribute > Delete.