Global Information Assurance Certification Paper

Windows 2000 Active Directory Schema

Securing Windows GCNT Practical Assignment V2.1b
Ian Parker
June 2001

This page intentionally left blank.

Table of Contents

Table of Figures

This page intentionally left blank.

Introduction

Active Directory is one of the key components of Windows 2000. In order to fully understand Active Directory, one must acquire an in depth knowledge of the component that forms the blueprint for the directory service, namely the schema. Furthermore, as the integrity of the schema is essential to the correct operation of a Windows 2000-based network, an understanding of how to effectively secure the schema against accidental or malicious damage is also essential.

This paper begins with an in depth look at the structure of the schema, discusses the utilities available for viewing the schema, walks through the process of implementing schema extensions and concludes by discussing the security safeguards that can be implemented to protect the schema from damage.

2. Structure of the Schema

The schema contains the definitions for all objects within a Windows 2000 forest. The schema consists of a set of classes and attributes that determine the kinds of objects that can be created in the directory and the properties of those objects. A particular class of objects share a set of common characteristics. Each class contains a set of mandatory attributes that all objects instantiated from that class must contain, as well as a set of optional attributes. Each attribute uses one of a pre-defined set of syntaxes that describes what type of data may be assigned to the attribute.

The base schema that ships with Windows 2000 contains all of the classes and attributes required by the operating system, a total of one hundred and forty two classes and eight hundred and sixty three attributes. However, the schema can be extended to include brand new classes or to add new attributes to existing classes. Schema extensions may be performed manually by administrators or by directory-enabled applications, such as Exchange 2000.

2.1 Classes

Classes provide templates for objects, such as users, computers and printers, that may be created in the directory and describe both the mandatory and optional attributes of these objects. Figures 1 through 3 show the properties sheets of a typical class, as displayed by the Active Directory Schema MMC snap-in.

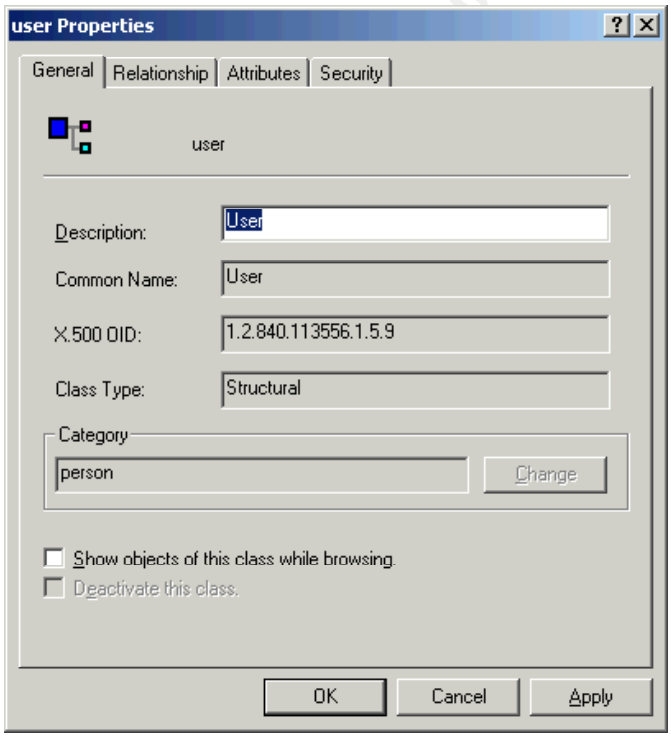

Figure 1 - Properties of a Class - General Tab

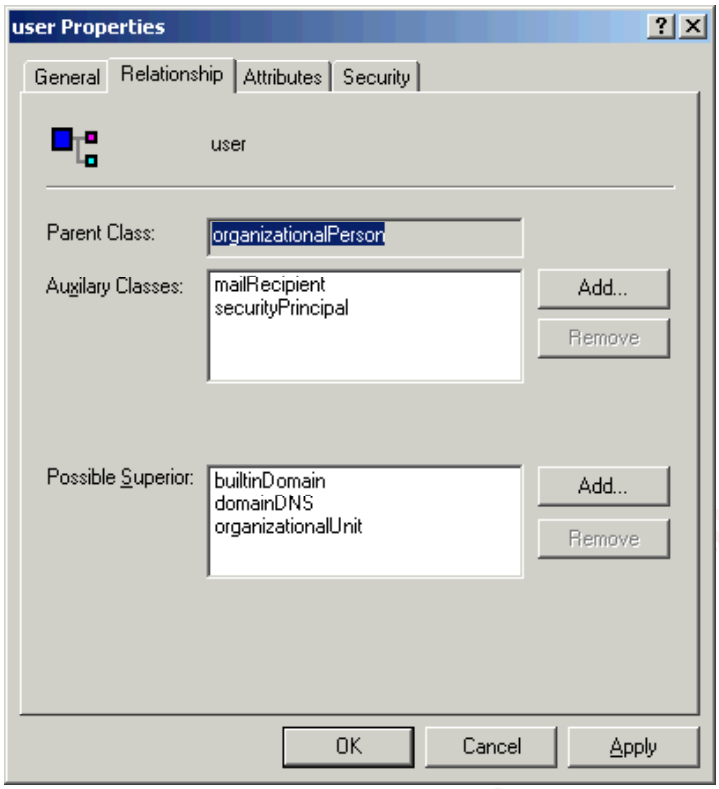

Figure 2 - Properties of a Class - Relationship Tab

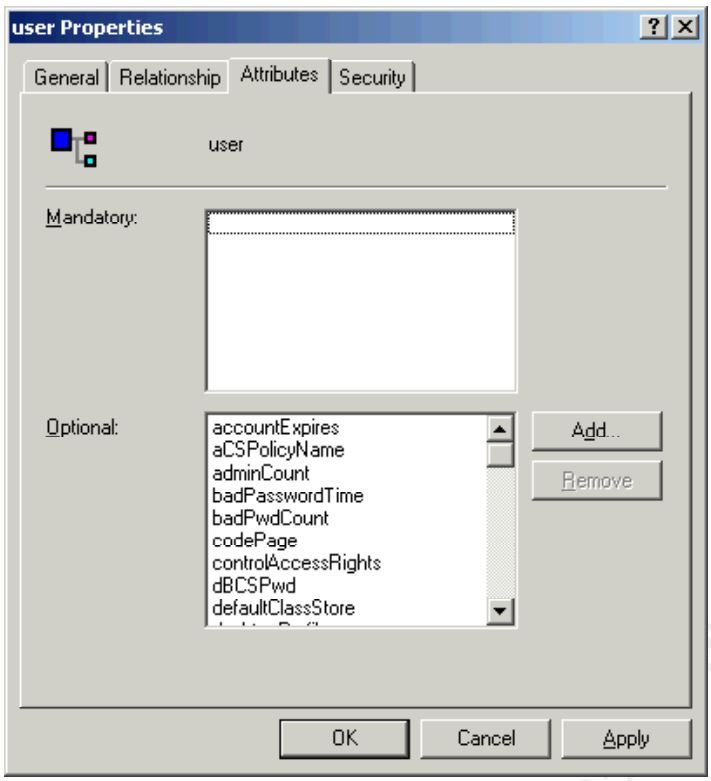

Figure 3 - Properties of a Class - Attributes Tab

A description of these properties follows.

2.1.1 Description

This is the description of the class that is displayed in the user interface.

2.1.2 Common Name

This is the name used to represent the object.

2.1.3 X.500 OID

X.500 is a set of standards for directory services. Active Directory is an "X.500-like" directory, although it only implements a subset of the X.500 standards. OIDs (Object Identifiers) are numeric values that uniquely identify every object in the directory. OIDs are implemented using a tree structure, similar to a DNS hierarchy. Figure 4 shows how the OIDs used by objects within Active Directory are derived.

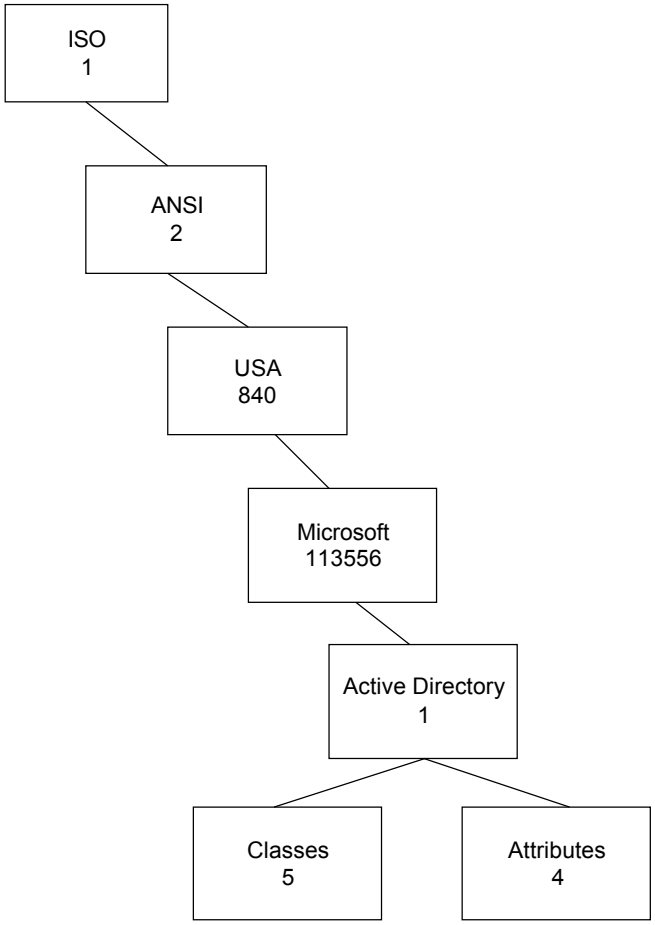

Figure 4 - OID Tree Structure

The ISO (International Standards Organization) is the root authority for all OIDs and has assigned the number 2 to ANSI (American National Standards Institute), which is the issuing authority within the United States. ANSI, in turn, has assigned the number 840 to companies within the United States and 113556 to Microsoft. Finally, Microsoft uses 1 for Active Directory and 5 and 4 for classes and attributes, respectively, below this level.

For example, notice from Figure 1 that the User class has an OID of 1.2.840.113556.1.5.9.

2.1.4 Class Type

Classes can be one of four types, namely, structural, abstract, auxiliary and 88.

Structural classes are the only classes that can have objects directly instantiated from them. For example, User objects are instantiated from the User class. Structural classes inherit attributes from both Abstract classes and other Structural classes. They can also include properties from Auxiliary classes.

Abstract classes cannot be used to instantiate objects directly. Rather, their purpose is to provide a template for the creation of Structural classes. Abstract classes inherit attributes from other Abstract classes.

Auxiliary classes contain a list of attributes that can be used in multiple Structural or Abstract classes. Auxiliary classes inherit attributes from other Auxiliary classes.

88 Classes are those that were defined prior to the X.500 1993 specification, which defined class types. New 88 classes should not be created.

Note that class types are also referred to as class categories. This is different from object categories, which are discussed next.

2.1.5 Object Category

The object category is used to group classes together to facilitate searches. Notice from Figure 1 that the User class is in the Person category. As we will discuss shortly, the Person class is a parent of the User class.

2.1.6 Parent Class

Classes are organized in a hierarchy. This is so that attributes can be defined once and used by multiple classes. A class propagates all attributes associated with it, both mandatory and optional attributes, to all classes that reference that class as a parent.

Figure 5 shows a small portion of the class hierarchy that relates to the User class. Notice that User is derived from Organizational-Person which, in turn, is derived from Person. Finally, Person is derived from Top which, as the name suggests, is at the very top of the class hierarchy. Notice also that Top, Person and Organizational- Person are all abstract classes, while User is a structural class. As discussed earlier, this means that the first three classes can only be used to derive other classes, while the User class can be used to instantiate an object, in this case, user objects.

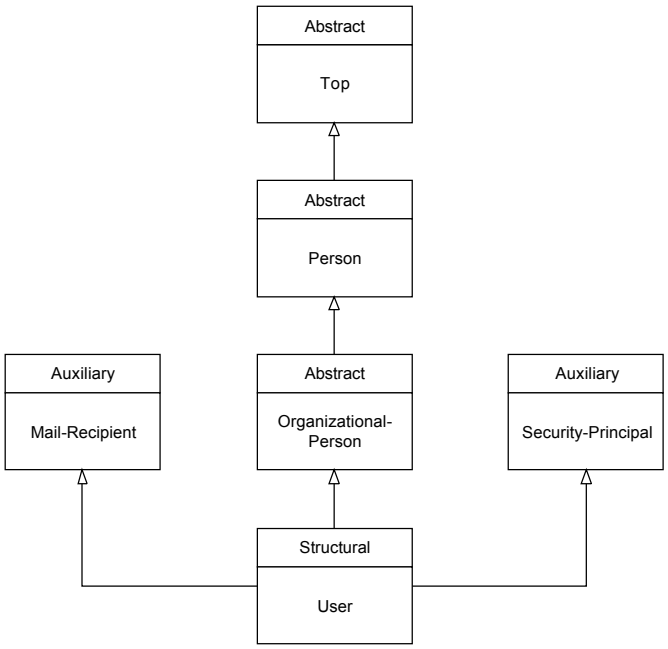

Figure 5 – Class Hierarchy for User Class

Top only has four mandatory attributes, but a great many optional attributes, all of which are propagated to every other class in the schema. Many of these attributes will have no relevance for certain classes. For example, the User class has many optional attributes that will never appear in dialog boxes related to configuration of user accounts.

2.1.7 Auxiliary Classes

Referring again to Figure 5, notice that User also inherits attributes from two auxiliary classes, Security-Principal and Mail-Recipient. As discussed earlier, auxiliary classes provide attributes that are typically used by multiple classes.

2.1.8 Possible Superior

This property specifies classes that can contain an instance of the current class. Notice from Figure 2 that the User class has three possible superiors, builtinDomain, domainDNS and organizationalUnit. This means that a User object can be contained within objects instantiated from these classes, i.e. the Built-in Domain container, a domain or an Organizational Unit.

2.1.9 Mandatory Attributes

Mandatory attributes must be given values before an instance of the object can be created. Referring to Figure 3, one might conclude that the User class has no

mandatory attributes. However, this is misleading, as the property sheet only shows attributes directly applied to this object. In fact, the User class derives four mandatory attributes from Top, one from Person, one from Mail-Recipient and two from Security-Principal. This can be seen from the Active Directory Schema console by selecting the User class in the scope pane. All of the attributes associated with that class are listed in the results pane, as shown in Figure 6.

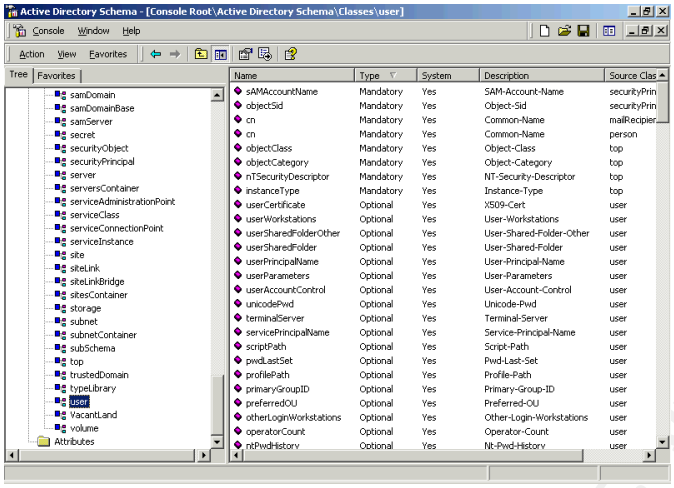

Figure 6 – Viewing All Attributes of a Class

2.1.10 Optional Attributes

Optional attributes are just as the name suggests. Again, the property sheet for a class only lists the optional attributes directly applied to that class. A class may have many optional attributes that have no relevance for the objects instantiated from that class. For example, the User class has two hundred and four optional attributes, many inherited from other classes.

2.2 Attributes

Attributes describe the classes defined in the schema. As we have discussed, a single attribute can be applied to many classes. Figure 7 shows the properties page of a typical attribute, as displayed by the Active Directory Schema MMC snap-in.

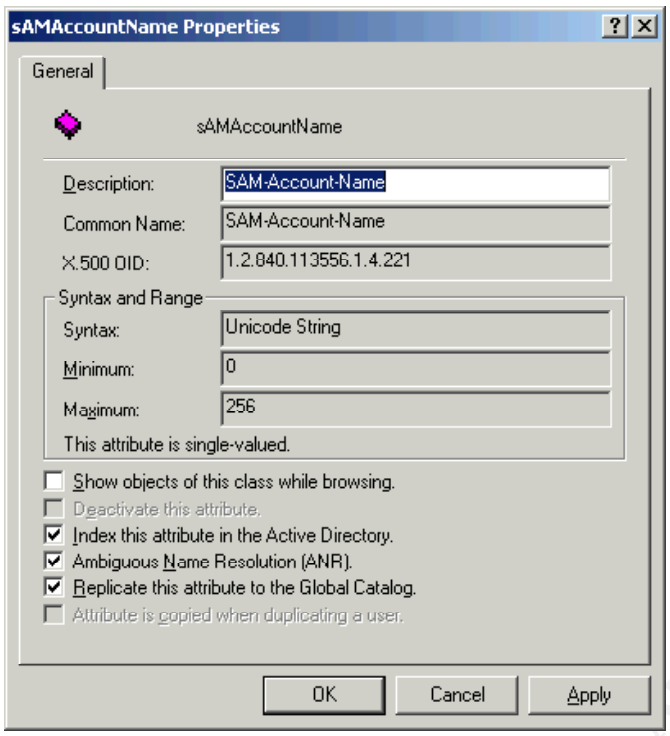

Figure 7 - Properties of an Attribute

The description, common name and X.500 OID are exactly as described for classes. A description of the other properties follows.

2.2.1 Syntax and Range

Syntax defines the type of data that the attribute will contain. Twenty three syntaxes are pre-defined and no new syntaxes can be created. The maximum and minimum values define the possible range of values.

2.2.2 Single -value and Multi-Value Attributes

Attributes can be single or multi-valued. The Lightweight Directory Access Protocol (LDAP) reads a multi-value attribute as a single entity. This can be problematic if the attribute contains a large number of values. A draft RFC proposes a mechanism for incremental retrieval of multi-valued attributes. Servers that support incremental retrieval include the OID 1.2.840.113556.1.4.802 in the supportedControl attribute on the rootDSE object. Notice from the LDP display shown in Figure 8 that Windows 2000 does not currently support this option.

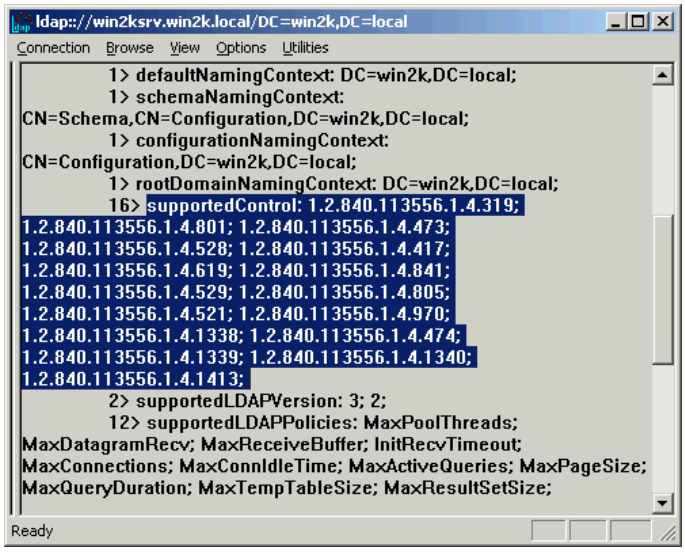

Figure 8 – Viewing supportedControl Attribute

2.1.3 Indexing

Indexing makes directory searches that include that attribute more efficient. Indexed attributes should be single value, highly unique and evenly distributed across all objects that contain them. Indexing multi-value attributes is not recommended, as this exacts a penalty both in performance and storage space. Indexing is enabled by checking the box shown in Figure 7.

2.1.4 Ambiguous Name Resolution

Ambiguous Name Resolution (ANR) is a search algorithm used in conjunction with indexing to facilitate searches. Its purpose is to enable clients to perform directory searches in cases where only part of the attribute's value is known.

The ANR check box is grayed out until indexing of the attribute is enabled. ANR can only be enabled for certain types of attributes, specifically Unicode or Teletex strings. By default, only nine attributes have this property set.

2.1.5 Replication to Global Catalog

The Global Catalog contains a copy of all objects within an Active Directory forest, but only a subset of their attributes. Its main purpose is to enable users to quickly search on often used attributes, without having to know the location of the object within the forest. An attribute is replicated to the Global Catalog by checking the box shown in Figure 7. Be aware that replication of too many attributes to the Global Catalog can have a large impact on network bandwidth throughout the forest.

3. Viewing the Schema

The schema may be viewed using several different tools. For security reasons, none of these tools are available out of the box, but must be installed by the administrator. Following is a brief description of how to install the tools and use them to view schema information.

3.1 Active Directory Schema Console

The most user-friendly tool that Microsoft provides for managing the schema is the Active Directory Schema MMC Snap-in. This is not one of the default snap-ins provided with the system, however. In order to add this snap-in to the list of available snap-ins, the administration tools package (adminpak.msi) must be installed. The DLL for the snap-in must then be registered by executing the command regsrv32 schmmgmt.dll from a command prompt or the Run command. The snap-in can now be added to a new or existing MMC console in the usual manner. Figure 9 shows the Active Directory Schema snap-in with the classes folder expanded to show some of the available classes.

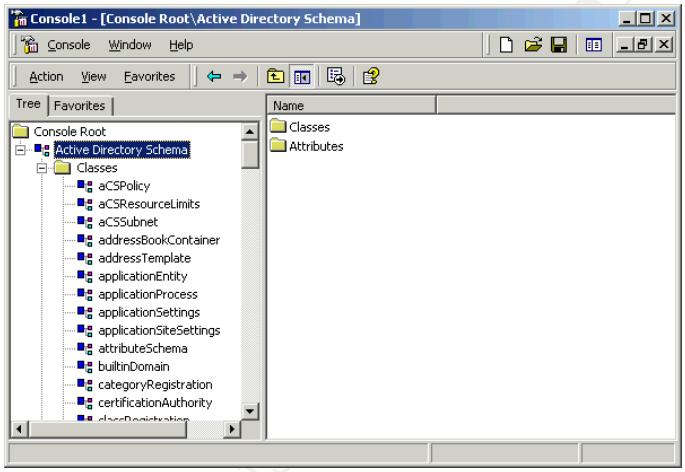

Figure 9 - Active Directory Schema Utility

3.2 ADSI Edit Console

While the Active Directory Schema snap-in displays a high level view of the schema, the other two tools display the raw data in a much more granular fashion. The first of these, ADSI Edit, is another MMC snap-in. However, the DLL associated with this utility must be installed from the Support/Tools folder on the Windows 2000 CD. Once the snap-in has been added to a console, the schema partition can be viewed using the following procedure:

- Right click ADSI Edit and select Connect to. A Connection dialog box will appear.
- In the Connection Point check box, ensure that Naming Context is selected.
- Select Schema from the Naming Context box and click OK.

Figure 10 shows the resulting display. Note that the schema container itself is an object and, like all other objects in Active Directory, has a Distinguished Name (DN). The schema container is contained within the configuration container which, in the example shown, resides in a forest root domain called win2k.local.

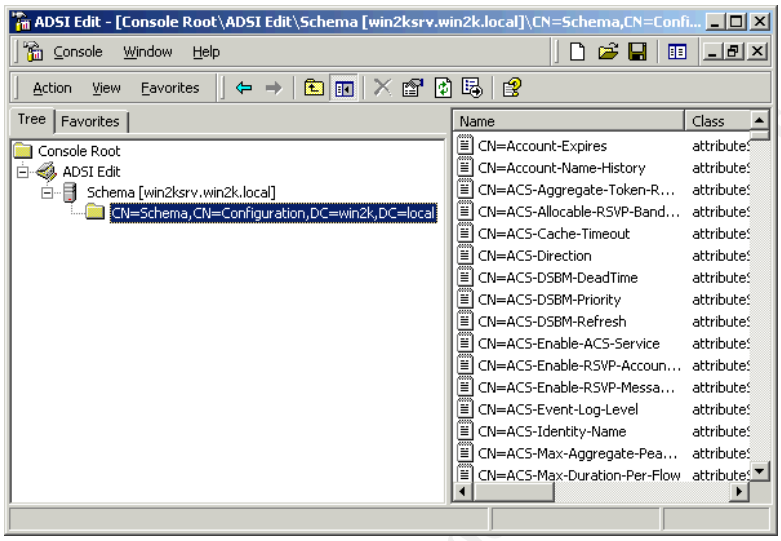

Figure 10 - ADSI Edit Utility

3.3 LDP Utility

The third tool, LDP, which must also be installed from the Support/Tools folder on the CD, is not an MMC snap-in but, instead, has its own GUI interface. LDP is a generalized LDAP query tool that can be used to search for information in other X.500-compliant directories besides Active Directory. This is the least user friendly of the three methods for viewing schema information, but presents the most detail. After starting the program, the schema may be viewed using the following procedure:

- Go to Connection | Bind.
- In the Bind dialog box, enter the username and password of a user who is a member of the Schema Admins group.
- Go to Tree | View
- In the Tree View dialog box, enter the DN of the schema and click OK.

Figure 11 shows the resulting display.

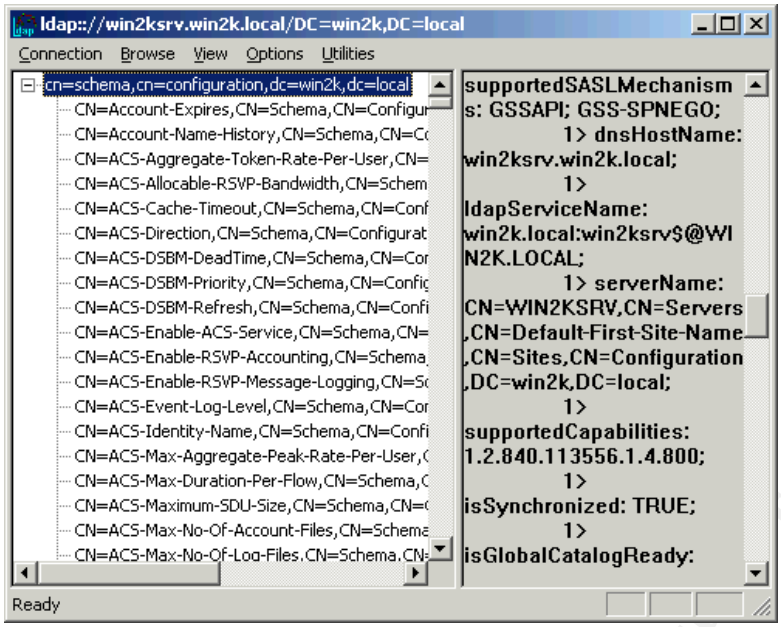

Figure 11 - LDP Utility

As a search tool, LDP can be used to find information that is not available using the other tools. For example, if you wish to find out which attributes are indexed, perform the following procedure:

- Start LDP.
- Bind to the domain of interest using the appropriate credentials.
- Go to Browse | Search and enter the DN of the schema in the Base DN box.
- Enter the following filter into the Filter Box:
  (&(objectCategory=attributeSchema)(searchFlags:1.2.840.113556.1.4.803:=1))
- Click Subtree for the search scope and click the Options button.
- Enter the attributes to be returned in the results, for example:
  objectClass;name;cn;distinguishedName;
- Click OK in the Search Options box and then click Run to execute the query.

The results appear in the LDP window, as shown in Figure 12, which indicates that sixty four attributes in this schema are indexed.

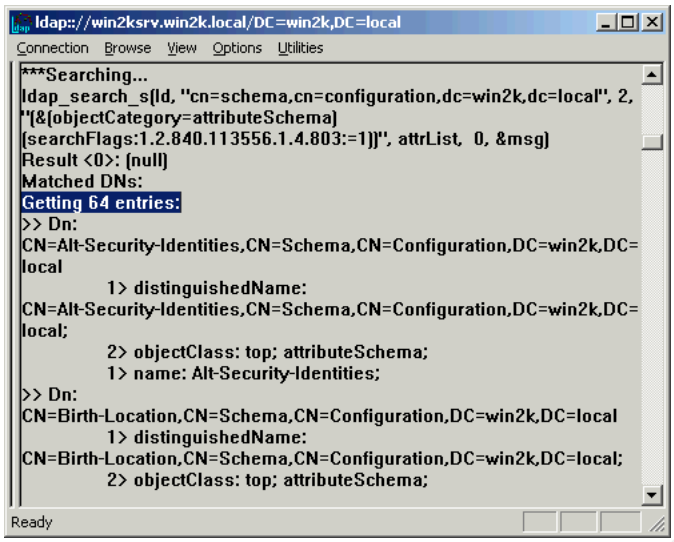

Figure 12 – Viewing Indexed Attributes Using LDP

A great deal of information can be obtained using this tool after one masters the rather obtuse syntax for specifying filters. For example, substituting the filter (&(objectCategory=attributeSchema)(isMemberOfPartialAttributeSet=TRUE)) into the above procedure will return the total number of attributes that are replicated to the Global Catalog (one hundred and thirty eight by default) and list those attributes, as shown in Figure 13.

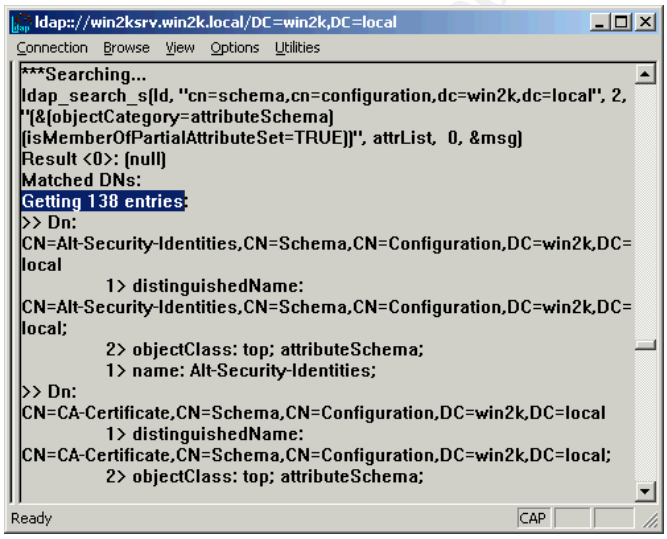

Figure 13- Viewing Attributes Replicated to Global Catalog using LDP

3.4 Finding the Schema

Directory-aware applications and scripts may sometimes need to locate information in the schema container without knowing the name of the domain in which it resides. This is known as serverless binding. This can be accomplished by binding to a special entry at the top of the logical namespace called rootDSE. An attribute of rootDSE called schemaNamingContext, provides the location of the schema.

To locate the schema from rootDSE using ADSI Edit, perform the following procedure:

- Open the ADSI Edit MMC.
- Right click ADSI Edit and select Connect to. A Connection dialog box will appear.
- In the Connection Point check box, ensure that Naming Context is selected.
- Select RootDSE from the Naming Context box and click OK.
- In the Console Tree expand the ADSI Edit node. The RootDSE node is displayed.
- Expand the RootDSE node. The RootDSE folder is displayed.
- Right click the RootDSE folder and select Properties.
- In the Select a Property to view dialog box, select schemaNamingContext from the list of properties. The DN of the schema will be shown under Attribute Values, as shown in Figure 14.

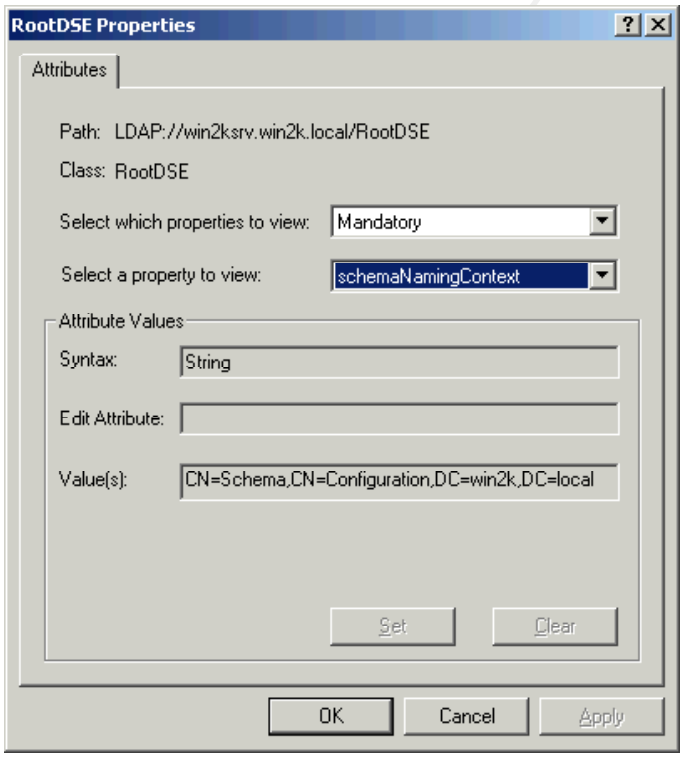

Figure 14 - Locating Schema using ADSI Edit

4. Extending the Schema

4.1 Why Extend the Schema

The schema may be extended either by modifying existing attributes or classes or by creating new attributes or classes. Schema extensions can be performed manually by administrators or automatically by applications. It is important to realize that Microsoft expects Active Directory to be an enterprise-wide repository for all kinds of information that currently lies in disparate, proprietary databases. As a result, many future applications will be so-called directory-aware applications i.e. they will use Active Directory to store their configuration data. Some of this data may require the creation of new classes or attributes. For this reason, it is important for administrators of a Windows 2000 network to understand what occurs when the schema is extended.

This said, Microsoft recommends that the schema only be extended when absolutely necessary. There are two reasons for this recommendation. First and foremost, extending the schema requires a high level of knowledge and mistakes can wreak havoc on the entire network. Secondly, schema objects cannot be deleted, so if you run a script that creates a hundred new attributes and then change your mind, you now have a hundred redundant objects in the directory. It is possible to deactivate these objects so that no new or existing classes can use them, but they still clutter up the database.

4.2 How To Extend The Schema

The basic steps for extending the schema are as follows:

- Make a backup! A system backup, including the System State information, should be made of a domain controller that has an up-to-date copy of the schema. In the event that schema corruption occurs as a result of your changes, this backup can be used to perform an authoritative restore. This can be a restore of the entire schema or, if the problem is limited in scope, to individual branches.

- Obtain an OID for each new class or attribute that you plan to create. If your Windows 2000 forest will be accessible from the Internet, these OIDs need to be globally unique and, therefore, need to be registered with the governing body within your country that issues OIDs. This is similar to the assignment of IP addresses. However, in the same way that private IP addresses can be used on internal networks, if your network will never be accessible from the Internet, you can use a Resource Kit utility called OIDGEN. When run from the command prompt, OIDGEN generates two base OIDs, by taking the OID

assigned to Microsoft and appending a GUID (Globally Unique Identifier). One base OID is generated for classes and one for attributes. OIDs for new classes and attributes can then be defined by adding a numeric value to the end of the base OID and incrementing the value for each object.

- Find out which domain controller is performing the role of Schema Master. This is the only domain controller in an Active Directory forest that holds a writeable copy of the schema, so your schema extensions must be made at this domain controller. To find out which domain controller is the Schema Master, perform the following:

    - Open the Active Directory Schema console.
    - Right click Active Directory Schema and select Operations Master.
    - The domain controller currently performing the role of Schema Master is displayed, as shown in Figure 15.

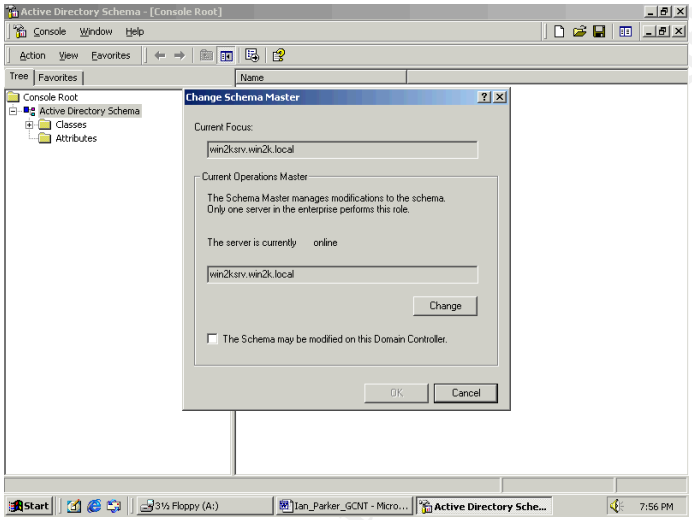

Figure 15 – Locating Schema Master

- Ensure that your account is a member of the Schema Admins group. This is a universal or global group (depending on whether the domain is in native or mixed mode) in the root domain of the forest. By default, only members of the Schema Admins group have permissions to modify the schema. To verify that your account has the required permissions, perform the following:

    - Open the Active Directory Users and Computers console.
    - Expand the domain node.
    - Open the Users folder.

- Right click the Schema Admins group in the results pane and select Properties.
- View the Members tab. The current members of the Schema Admins group will be displayed, as shown in Figure 16. By default, only the domain administrator account is a member of this group. Other users may be added if necessary.

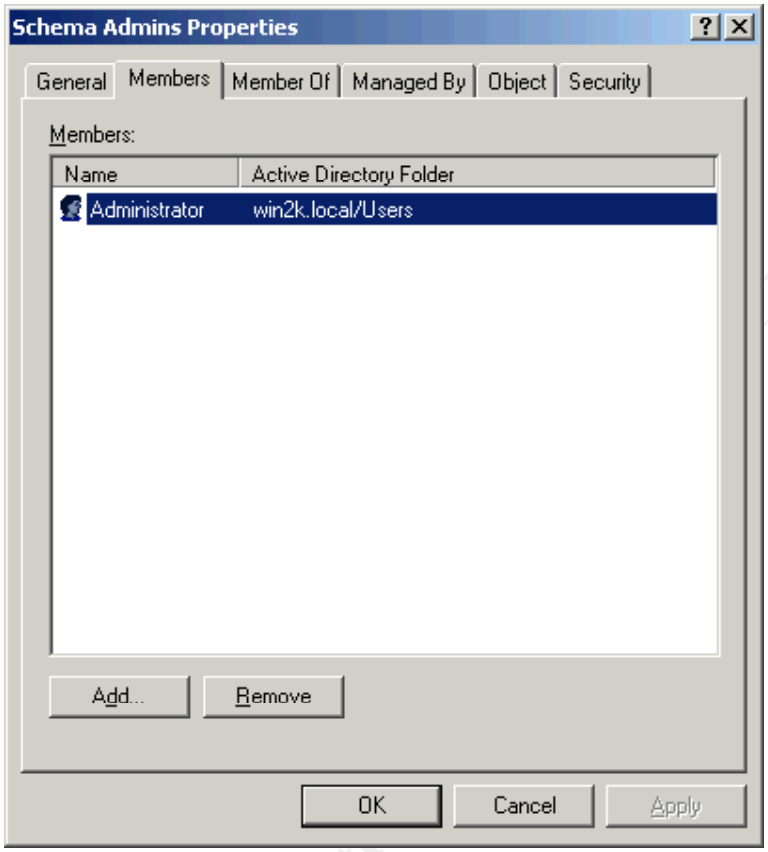

Figure 16 – Viewing Members of Schema Admins Group

- Enable write access to the schema. By default, access to the schema, even for members of the Schema Admins group, is read only. A registry entry on the Schema Master domain controller must be changed to enable write access. Microsoft refers to this as the safety interlock. The registry can be modified directly be adding an entry of type REG_DWORD called Schema Update Allowed to the key HKLM\System\CurrentControlSet\Services\NTDS\Parameters and setting its value to 1. Otherwise, it can be changed via the Change Schema Master screen shown in Figure 15, by checking the box labeled "The Schema may be modified on this Domain Controller".

- Create the desired new attributes.

- Either wait five minutes for the schema cache to be updated, trigger a cache reload or, in the following step, reference the new attributes by OID. To trigger a cache reload, perform the following:

  - Open the Active Directory Schema console.
  - Right click Active Directory Schema and select Reload the Schema, as shown in Figure 17.

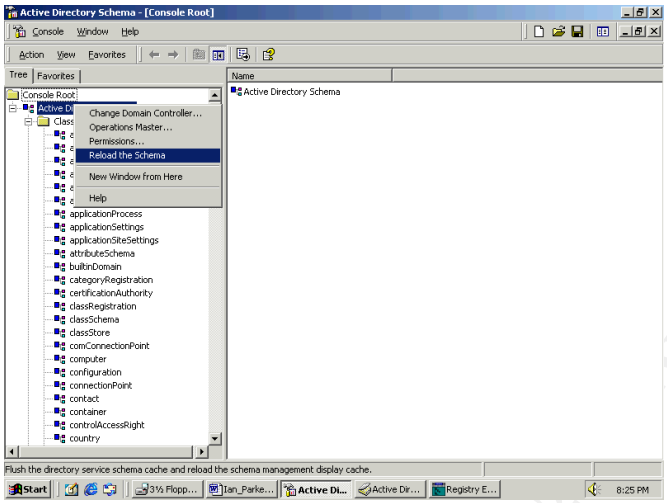

Figure 17 – Reloading Schema

- Create the desired new classes and add the new attributes. If the new classes will be used within five minutes, another cache reload is required.

- Re-activate the safety interlock by setting the registry entry added earlier to 0.

- Document all schema changes, using a tool such as SchemaDoc provided by Microsoft.

4.3 Schema Extension Example

Let's walk through a simple example to see how this works in practice. We will assume that the pre-requisite steps have been performed i.e. we have obtained base OIDs for our new attributes and classes (in this case generated by OIDGEN), we have determined which domain controller is the Schema Master, we have verified that we have permissions to modify the schema and we have removed the safety interlock.

Let's say that we want to use Active Directory to store information on real estate

properties for sale. We might create a class called Properties, which would include attributes that are common to all types of real estate. One such attribute would be the price of the property. We could then create subclasses for each type of property, for example, apartment buildings, townhouses, detached homes, vacant land etc. Each of these subclasses would have attributes that are relevant for that type of class. The actual properties would be instances of these subclasses.

A class hierarchy that would meet our needs is illustrated in Figure 18. In this example, the Properties class is a subclass of Top and is an abstract class. The subclasses of Properties are structural classes.

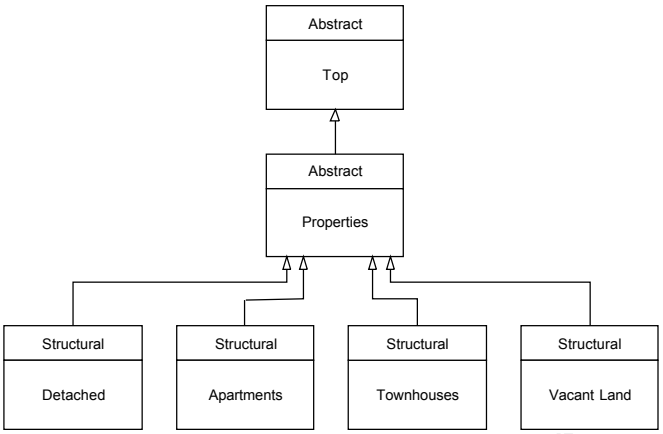

Figure 18 – Class Hierarchy for Real Estate Example

First, we create the required attributes. To create new attributes, we perform the following steps:

- Open the Active Directory Schema console and expand the Active Directory Schema container.
- Right click on the Attributes folder and select Create Attribute. A warning message will appear to remind you that creation of objects is irreversible.
- Click continue and the Create New Attribute dialog box will appear.
- Enter the information for the attribute.

Figure 19 shows the Create New Attribute dialog box filled in for an attribute called Price. Other attributes would be created in a similar manner.

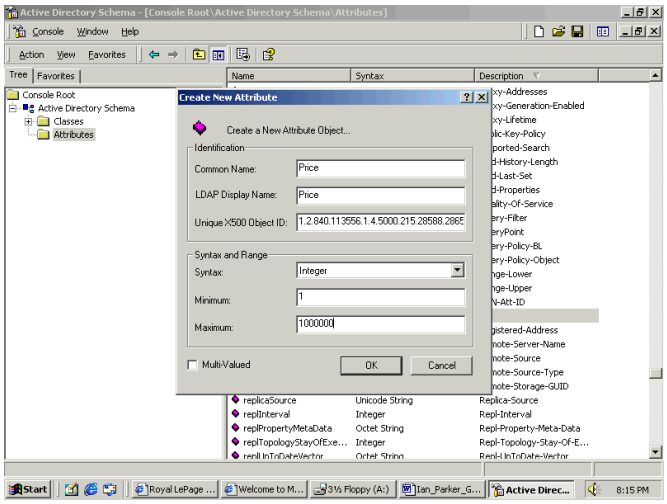

Figure 19 – Creating a New Attribute

We now proceed to create our new classes. To create a class, perform the following steps:

- Open the Active Directory Schema console and expand the Active Directory Schema container.
- Right click on the Classes folder and select Create Class.
- Move past the warning dialog box and the Create New Schema Class dialog box will appear.
- Enter the information for the class.

Figures 20 and 21 show the completed information for the Properties class. Other classes would be created in the same manner.

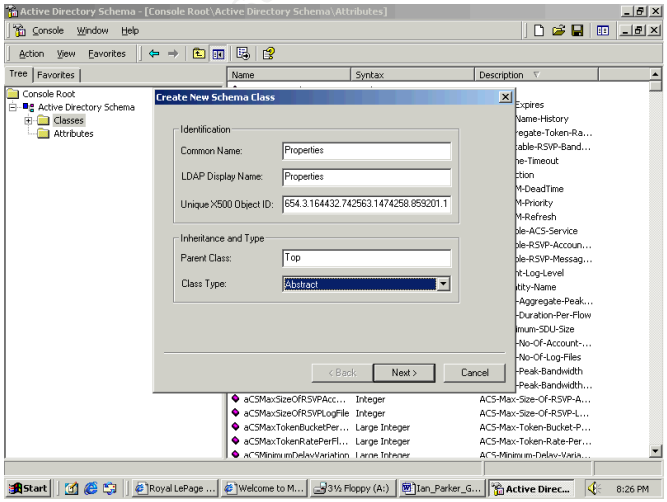

Figure 20 – Creating New Class, First Screen

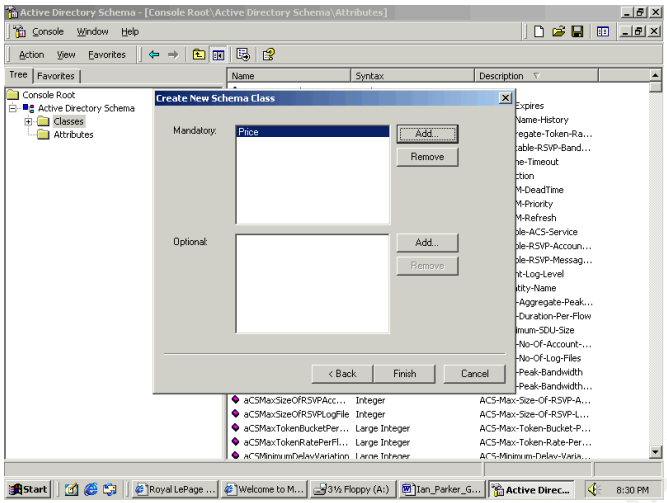

Figure 21 – Creating New Class – Second Screen

We can now return to the Active Directory Schema console and display the Properties sheet for each new object to complete the Description field. Figure 22 shows the Properties sheet for an attribute called Bedrooms, whose value is the number of bedrooms. Figure 23 shows how the attribute now appears in the results pane of the Active Directory Schema console.

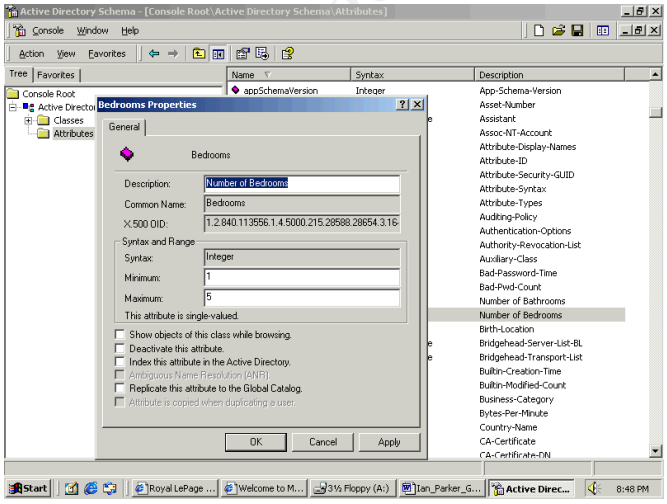

Figure 22 – Entering Description of Attribute

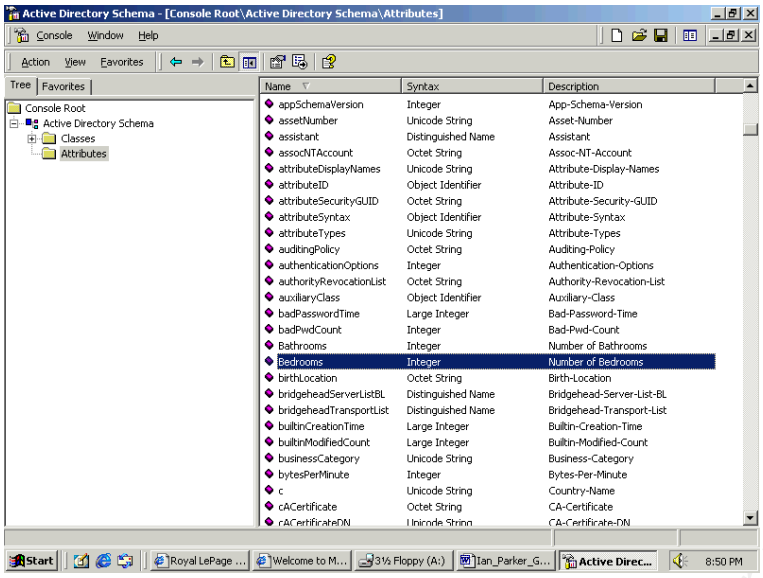

Figure 23 – New Attribute in Active Directory Schema Console

5. Schema Security

The security of the schema is critical to the reliable operation of a Windows 2000 network. We have already discussed some of the safeguards that are built in to the operating system to protect the schema from accidental or malicious damage. Following are some suggestions on how these safeguards may be extended to provide a truly secure environment for this critical component of Windows 2000.

5.1 Secure the Schema Master

Only one domain controller, the Schema Master, contains a writeable copy of the schema. This is a great benefit from a security standpoint, as security efforts can be focused on this system. Obviously, the Schema Master should be in a physically secure location and both local and network access should be highly restricted. A discussion of how to secure a Windows 2000 system is beyond the scope of this document. Many excellent sources may be used to assist in this endeavor, including the SANS document listed in the References section.

5.2 Restrict Access to Schema Management Tools

Securing the Schema Master, however, does not by itself solve all security issues. The role of Schema Master can be transferred to another domain controller using the Active Directory Schema console or the NTDSUTIL utility. The role can also be seized using NTDSUTIL in the event that the current Schema Master has crashed. Tight control of all schema management tools is essential both to prevent

unauthorized schema modification and to ensure that the Schema Master role is not purposefully transferred to a less secure system. The tools should either be removed entirely from the network when not in use or highly restrictive ACLs should be employed, so that only a select few individuals can use them.

5.3 Restrict Membership in the Schema Admins Group

By default, only the domain administrator belongs to the Schema Admins group. Membership in this group should be very restricted. In fact, the group could be left empty until such time as a schema modification is required. One weakness of this approach is that anybody who is a domain administrator in the forest root domain can add their account to the Schema Admins group. Some organizations have eliminated this weakness by using a phantom root domain that only contains the default, built-in administrative accounts and groups. All other user accounts, groups and computers are placed in one or more child domains. Of course, this approach incurs the hardware cost of additional domain controllers, as well as a greater administrative overhead. This solution, therefore, may not be appropriate for smaller organizations or those on a tight budget.

5.4 Remove Schema Modify Permissions

One step beyond leaving the Schema Admins group empty is to remove from this group the right to modify the schema. This can be done using the following procedure:

- Open the Active Directory Schema console.
- Remove the safety interlock.
- Right click Active Directory Schema in the scope pane and select Permissions.
- Click the Change Schema Master Permission button.
- Modify the permissions for the Schema Admins group, as shown in Figure 24. Removal of Write and Create All Child Objects permissions will prevent schema modifications.

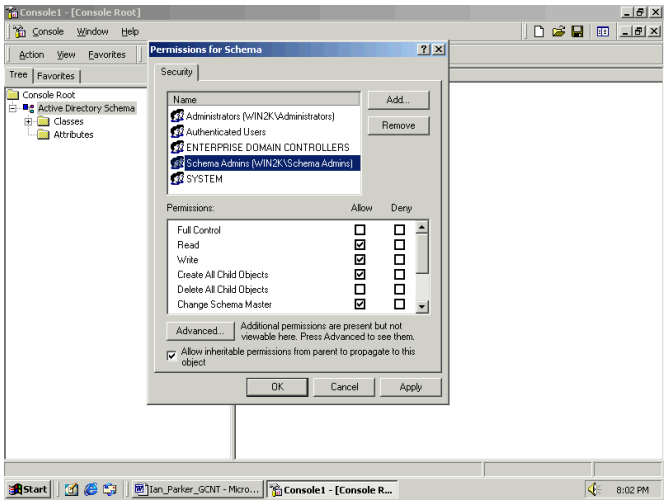

Figure 24 - Modifying Schema Permissions

- Re-apply the safety interlock.

5.5 Apply and Monitor the Safety Interlock

The safety interlock should be re-applied after every schema change. The registry key that applies the interlock should be tightly secured and auditing placed on any access to the key. If a host-based intrusion detection system is being used, it should also be watching for unauthorized access to this key.

5.6 Enable Auditing of Directory Service Access

By default, auditing of both successful and failed attempts to write to the schema is set for the Everyone group, as shown in Figure 25.

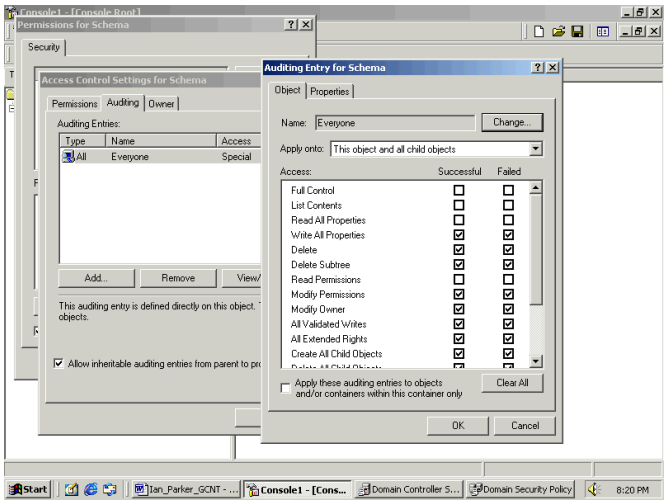

Figure 25 - Auditing of Schema Access

However, as is the case with auditing of file and printer access, the audit policy must first be enabled using Group Policy. This is performed by enabling success and failure auditing for the Audit Directory Service Access policy.

6. References

1. Books

   - Microsoft Corporation. Microsoft Windows 2000 Server Resource Kit. Microsoft Press, 2000. Microsoft Windows 2000 Server Distributed Systems Guide. Part 1 - Active Directory. Chapter 4 - Active Directory Schema.

   - SANS Institute. Securing Windows 2000 Step By Step. 2001

2. Microsoft Knowledge Base Articles

   - Q216060 - "Registry Modification Required to Allow Write Operations to Schema." Last Reviewed: November 2, 2000. URL: http://support.microsoft.com/support/kb/articles/Q216/0/60.asp

   - Q219005 - "Windows 2000: LDAPv3 RootDSE." Last Reviewed: December 31, 1999. URL: http://support.microsoft.com/support/kb/articles/Q219/0/05.asp

   - Q224543 - "Using Ldp.exe to Find Data in the Active Directory." Last Reviewed: October 27, 2000. URL: http://support.microsoft.com/support/kb/articles/Q224/5/43.asp

- Q228776 - "Setting User Rights for Designating FSMO Roles in an Enterprise." Last Reviewed: December 31, 1999. URL: http://support.microsoft.com/support/kb/articles/Q228/7/76.asp

- Q229662 - "How to Control What Data Is Stored in the Global Catalog." Last Reviewed: December 31, 1999. URL: http://support.microsoft.com/support/kb/articles/Q229/6/62.asp

- Q229691 – "How to Enable Domain Controllers to Modify the Schema." Last Reviewed: December 31, 1999. URL: http://support.microsoft.com/support/kb/articles/Q229/6/91.asp

- Q230662 - "Enumerating Indexed Attributes in Windows 2000 Active Directory." Last Reviewed: December 31, 1999. URL: http://support.microsoft.com/support/kb/articles/Q230/6/62.asp

- Q230663 - " How to Enumerate Attributes Replicated to the Global Catalog." Last Reviewed: May 12, 2000. URL: http://support.microsoft.com/support/kb/articles/Q230/6/63.asp

- Q232517 – "Global Catalog Attributes and Replication Properties." Last Reviewed: January 1, 2000. URL: http://support.microsoft.com/support/kb/articles/Q232/5/17.asp

- Q234790 – " How to Find FSMO Role Holders (Servers)." Last Reviewed: October 21, 2000. URL: http://support.microsoft.com/support/kb/articles/Q234/7/90.asp

- Q243299 – "Ambiguous Name Resolution for LDAP in Windows 2000." Last Reviewed: December 31, 1999. URL: http://support.microsoft.com/support/kb/articles/Q243/2/99.asp

- Q243311 – "Setting an Attribute's searchFlags Property to Be Indexed for ANR." Last Reviewed: December 24, 2000. URL: http://support.microsoft.com/support/kb/articles/Q243/3/11.asp

- Q255690 – "How to View and Transfer FSMO Roles in the Graphical User Interface." Last Reviewed: March 29, 2000. URL: http://support.microsoft.com/support/kb/articles/Q255/6/90.asp

- Q256938 – "Default Global Catalog Attributes in Windows 2000 Active Directory Schema." Last Reviewed: May 7, 2000. URL: http://support.microsoft.com/support/kb/articles/Q256/9/38.asp

- Q257203 – "Common Default Attributes Set for Active Directory and Global Catalog." Last Reviewed: May 7, 2000. URL: http://support.microsoft.com/support/kb/articles/Q257/2/03.asp

- Q257218 – "Default Active Directory Attributes in the Windows 2000 Schema." Last Reviewed: March 29, 2000. URL: http://support.microsoft.com/support/kb/articles/Q257/2/18.asp

- Q285172 – "Schema Updates Require Write Access to Schema in Active Directory." Last Reviewed: January 26, 2001. URL: http://support.microsoft.com/support/kb/articles/Q285/1/72.asp

3. Web Sites

- Lucent Technologies. "Windows 2000 Active Directory Design - Dedicated Forest Root." 2001. URL: http://www.ins.com/knowledge/whitepapers/win2kad.asp

- Microsoft Corporation. "Schema Documentation Program." Last Updated: February 3, 2000. URL: http://www.microsoft.com/TechNet/win2000/schema.asp

- Daily, Sean and Mar-Elia, Darren. "Chapter 1: Managing the Active Directory." The Definitive Guide to Windows 2000 Administration. URL: http://www.fastlane.com/windows2000admin/toc.cfm

- Microsoft Corporation. "Using Generated OIDs." December 5, 2000. URL: http://msdn.microsoft.com/library/psdk/adsi/glschemex_60s3.htm