IBM Vault Registry

Using the eNetwork X.500 Directory With IBM Vault Registry

Contents

Chapter 1. Overview

In prior releases of IBM® Vault Registry, the IBM eNetwork™ X.500 Directory was required. While with Vault Registry version 2.2.2, the X.500 Directory is no longer required, it is still supported. This document discusses how to set up the X.500 Directory server for use in a Vault Registry environment. It also provides guidelines for monitoring, backing up, and restoring the X.500 Directory.

Assumptions and Scope

This document covers information needed for integrating the eNetwork X.500 Directory with Vault Registry. Monitoring and backup and recovery are covered only at a high level.

This document is intended for operators and system administrators with a working knowledge of AIX and UNIX. It is strongly recommended that the document be read along with the chapter on setting up corequisite products in the *Installation and Configuration Guide*. In addition, it should be read within the context of your organization's schema documentation.

The eNetwork X.500 Directory, referred to here as the X.500 Directory or the Directory, uses the Stream Directory User Agent (sdua) scripting language. While a familiarity with sdua would be helpful, it is not required for use with this document. This document is intended to provide enough guidance for you to install and set up a standard implementation of the X.500 Directory without a working knowledge of sdua or schema configurations. For a further understanding of the sdua scripting language itself, refer to the *IBM DSSeries500 X.500 Electronic Directory System*. To understand the required underlying information for the appropriate use of the sdua scripting language and the X.500 schema configuration, refer to the X.500 standards documentation.

This document supports only a default installation and configuration of Vault Registry version 2.2.2.

About the Directory

IBM Vault Registry uses a Directory component to store information that can be used by your registration and certification validation applications. For example, the Directory stores:

- Public key certificates – required for encryption verification and Secure Sockets Layer (SSL) connections
- The user attributes associated with a Distinguished Name (DN) profile – the owner's roles and privileges
- Certificate Revocation Lists (CRLs) – a list of the serial numbers for all revoked certificates

Vault Registry uses the Lightweight Directory Access Protocol (LDAP) to access and update your organization's Directory.

Your organization has chosen the IBM eNetwork X.500 Directory for use with Vault Registry. X.500 is a collection of nine International Standards Organization (ISO) standards which are also International Telecommunications Union (ITU)

recommendations. It specifies the models for storing data. It also specifies the protocols for retrieving data and for the interworkings of a collection of servers, each of which holds only a portion of the collective (distributed) data. The X.500 Directory can be better understood as a specialized distributed database, whose stored data or information has both an internal and external structure.

X.500 Directory Structure

Typical of Directories in general, each entry in the eNetwork X.500 Directory represents a single object, such as a person, organization, or device. The entry contains a set of attributes that help to uniquely identify the object, describe its properties, and delineate its privileges. The attributes can include a person's country of origin, organization, organizational unit, and common name, which is the name the person is commonly known by.

All Directory entries are organized into a hierarchical structure called the Directory Information Tree (DIT). This structure has a single root and an unlimited number nodes branching off from the root. Each node corresponds to a Directory entry identified by a distinguishing naming attribute. The DN for an entry is formed by joining the DN of its immediately superior entry with its relative distinguished name (RDN). The RDN, which is an attribute value assertion (AVA) or set of attribute value assertions (MAVA), is used to uniquely identify (or distinguish) an entry from other entries that are all immediate subordinate entries of the same superior entry.

A typical Directory hierarchy is structured as follows:

- The first node from the root usually contains an entry for each country in the world (`c`).
- The immediately subordinate node usually represents the organizations within a given country (`o`).
- The next subordinate node can contain unique entries for the divisions within an organization (`ou`).
- The next subordinate node can represent individuals identified by a common name (cn) and other distinguishing attributes such as an email address (`emailAddress`).

Expressing the DN of an entry for this sample Directory hierarchy depends on the syntax requirements of the Directory client accessing the entry. If the Directory client is the `sdua` Directory administration tool, a sample DN for the entry is expressed as:

```
/ c "US" / o "IBM" / ou "Registry" / cn "Chris Smith", emailAddress "cjsmith@us.ibm.com"
```

The LDAP can also be used as a Directory client access protocol for the IBM eNetwork X.500 Directory. Expressing the DN of the same entry for an LDAP Directory client is as follows:

```
cn=Chris Smith + emailAddress=cjsmith@us.ibm.com,ou=Registry,o=IBM,c=US
```

X.500 Directory Schema

The Directory schema is the Directory's set of rules for controlling the information that may be placed in the Directory. This document describes the X.500 Directory schema components as they relate to Vault Registry. Any X.500 Directory schema consists of the following components:

- Object classes (`objectClasses`)

- Matching rules (matchingRules)
- Attribute types (attributeTypes)
- Name forms (nameForms)
- Structure rules (dITStuctureRules)
- Content rules (dITContentRules)

These schema components are described here either because they have definitions that are specific to Vault Registry or their standard definitions have been enhanced.

The method for loading the schema at a given Directory node is up to you. This document addresses the method whereby you develop schema template files for each node type (such as an organization object class). The template files include the sdua definition of the schema to be loaded that is appropriate for that node type.

Note: Be sure to study the default database and associated schema definitions that come with the product before applying the changes suggested in this document.

Security Considerations

There are certain performance and security advantages to setting up the X.500 Directory on a machine separate from your IBM Vault Registry server. Regardless of where you install the Directory software, be sure to configure it behind the protection of your server's firewall.

The strong encryption features provided in Vault Registry do not necessarily extend to the encryption support provided through your Directory server program. Therefore, you should ensure that access to the Directory is limited to trusted, vault-based transactions that occur behind the firewall. Doing so is critical to protecting the integrity of your Directory entries.

Installation and Configuration Roadmap

To install and configure the X.500 Directory, you must perform the following tasks:

- Install the X.500 Directory software.
- Prepare the X.500 Directory before you configure it for use with the Vault Registry system.
- Set up the X.500 Directory schema.
- Configure the X.500 Directory entries for use with Vault Registry.

Chapter 2. Installing the X.500 Directory Software

You can install the X.500 Directory software on the same machine where you plan to install the Vault Registry server components or on a separate machine.

To install the X.500 Directory, you must explicitly mount the CD-ROM file system and provide a pointer to the directory containing the X.500 Directory installation images. Use the following procedure to install the Directory software:

1. Create a UNIX account for the X.500 Directory Administrator, such as `x500usr`, and set the password.
2. Log in to the AIX server as `root`.
3. Install the X.500 Directory server LLP by following these steps:
   a. Mount the X.500 Directory CD and go to the `/x500_server` directory as follows:

      ```
      mount -v cdrfs -r /dev/cd0 /mnt
      ```

      where `/dev/cd0` represents the name of your CD-ROM device, and `/mnt` represents the directory name where you want to mount the CD-ROM.
   b. Issue the following command to install the X.500 Directory server:

      ```
      installp -agXd /mnt/x500_server -e /tmp/install.log x500.server
      ```

      where `/mnt` represents the name of the directory where the CD-ROM is mounted and `/tmp/install.log` represents the filename where you want output from the `installp` command to be saved.

      This step installs the X.500 Directory server files under the `/usr/x500` directory. These files require about 50 MB of space. Note that `installp` automatically expands file systems when necessary when `-X` is used. After installation, use the `x500.setup` command to create one or more instances of the Directory System Agent (DSA), as described in the following steps.
   c. Log out and log in as the Directory Administrator (for example, `x500usr`).
   d. Create an instance of a DSA, under the Directory Administrator account.

      To do this, run the following setup script:

      ```
      x500.setup -d x500 -p profile
      ```

      where `x500` represents the full or relative pathname of a directory that you want to be the home directory of the X.500 Directory and *`profile`* is the location of the `.profile` file for the Directory Administrator.

      The home directory of the X.500 Directory is referenced throughout this document as `${VFHOME}`. If the above command is executed from the Directory Administrator's `${HOME}` directory, then `${HOME}/x500` is created as the X.500 home directory. Some X.500 Directory configuration and data files are copied to this directory. Softlinks are created for other binaries and executables that were installed under `/usr/x500`. The setup script also allows you to choose whether or not you want your user profile updated. The X.500 home directory requires about 15 MB of space for the basic setup of configuration and data files, and for runtime files (for example, logs). Add sufficient space for the database.

e. Establish the Directory Administrator's environment by logging out and logging in as the Directory Administrator (for example, x500usr).

f. Start the DSA, by entering the following command:

```
x500.run    /* to setup the environment and start the DSA */
```

This command starts the following processes:

- boing
- dbm
- pmslave
- dsa
- dot -b

For detailed instructions on installing the eNetwork X.500 Directory, refer to the *IBM DSSeries500 X.500 Electronic Directory System*.

Chapter 3. Preparing the System

To prepare your organization's X.500 Directory for use in a Vault Registry environment, you must perform the following tasks described in this chapter:

1. Set up the commsrv executable.
2. Specify the DSA setup options.
3. Set the DSA runtime options.
4. Set up the SSL links.

Setting Up the commsrv Executable

Vault Registry uses port 389 for the LDAP protocol. Since the number of this port is less than 1000, to set up the commsrv executable, make the following changes to it:

1. Log in as root.
2. Change the commsrv permissions by entering:

   ```
   chmod +s /usr/x500/bin/commsrv
   ```
3. Change the commsrv ownership by entering:

   ```
   chown root /usr/x500/bin/commsrv
   ```

Specifying the DSA Setup Options

Modify the DSA setup configuration file to specify DSA setup options, as follows:

1. Log in as the Directory Administrator, for example, x500usr.
2. Edit the $VFHOME/setup/config file.
3. Ensure that port 389 is used for LDAP communications by adding the following declaration to the configuration file:

   ```
   ldapsrvaddress = localhost:389
   ```
4. Declare a save directory by making the following entry:

   ```
   save = save
   ```

   The save directory is not a Vault Registry requirement. This file system directory is used by the sdua save command. The save command takes a snapshot of the Directory database even when the add and modify capabilities are available. The snapshot is stored as a set of files in the directory $VFHOME/save.
5. Make the declaration that enables commsrv to return binary attributes in ASCII hex by entering:

   ```
   ldapasciihex = on
   ```

Setting the DSA Runtime Options

The DSA control program (dsac) is a textual interface used to set the DSA runtime options. To set the DSA runtime options, follow these steps:

1. Log in as the Directory Administrator (x500usr).
2. At the AIX prompt, invoke the interactive interface by entering:

   ```
   dsac -g
   ```

3. At the dsac prompt, set and save server configuration parameters to the suggested value as shown here. Each command uses this syntax:

   `setwrite parameter = value`

   a. To set the option for Directory operation thread (dot) processes to the suggested value of 3, enter:

      setwrite dots=3

   b. To set the option for maximum dot size, in megabytes, to the suggested value of 20, enter:

      setwrite dotsize=20

   c. To set the option for maximum sessions to the suggested value of -1, enter:

      setwrite sessions=-1

   d. To set the option for maximum updates to the suggested value of 1, enter:

      setwrite updates=1

   e. To set the option for size limit to the suggested value of 2000, enter:

      setwrite sizelimit=2000

   f. To set the option for time limit to the suggested value of 120, enter:

      setwrite timelimit=120

   g. To set the option for cache size, in megabytes, to the suggested value of 128, enter the following commands:

      close
      setwrite cache=128
      open

   h. To set the option for Directory Access Protocol (DAP) timeout to the suggested value of 0, enter:

      setwrite daptimeout=0

   i. To set the option for Directory Server Protocol (DSP) timeout to the suggested value of 0, enter:

      setwrite dsptimeout=0

   j. To set the optimistic option to the suggested value of *on*, enter:

      setwrite optimistic=on

   k. To set the async mode option to the suggested value of *on*, enter:

      setwrite async=on

   l. To set the option for recovery to the suggested value of *on*, enter:

      setwrite recovery=on

   m. To set the option for enabling LDAP to the suggested value of *on*, enter the following command:

      setwrite ldap=on

      When this command completes, hit enter to return to the dsac prompt.

   n. To set the enable web option to the suggested value of *off*, enter:

      setwrite web=off

   o. To set the option for query logging to the suggested value of *on*, enter:

      setwrite qlog=on

   p. To set the option for update logging to the suggested value of *on*, enter:

      setwrite ulog=on

   q. To set the option for activity logging to the suggested value of *off*, enter:

      setwrite alog=off

   r. To set the option for LDAP connections logging to the suggested value of *on*, enter:

```
setwrite clog=on
```

s. To set the option for SEP size factor to the suggested value of 5, enter:

```
setwrite sizefactor=5
```

t. To set the option for search aliases to the suggested value of *on*, enter:

```
setwrite searchalias=on
```

u. To set the option for save and scratch to the suggested values of 20 and 4 respectively, enter the following commands:

```
close
init 20,4
open
```

4. To exit dsac, enter:

```
quit
```

5. To establish anonymous access, enter:

```
sdua -gd /usr/x500/setup/anonymous.access
```

6. To stop the Directory, enter:

```
x500.run stop
```

7. To start the Directory again, enter:

```
x500.run
```

Note: Ensure that the timelimit, sizelimit, and sizefactor parameters are set properly, because they directly affect the client sessions.

Setting Up the SSL Links

The eNetwork X.500 Directory typically requires the links to X.500 Directory libraries shown in Table 1. To determine whether the required links exist on your system, you can use one or both of the following methods:

- If you are having problems starting the commsrv process (LDAP to DAP converter and multiplexor), view the $VFHOME/general/error log.

  Look for a message generated by the commsrv process that indicates whether any of the links are unavailable. If you need to increase the level of detail displayed in the messages, you can edit the /$VFHOME/setup/config file and change the value for the debug switch.

- If commsrv is having problems accessing the library links, check the libraries to determine which links commsrv should be accessing and which links commsrv is not able to access. To do this, use the following command:

  ```
  dump -H commsrv
  ```

Table 1. SSL Links

Source File (to link)	Target File (linked to source file)
/usr/x500/lib/libvfssl.a	/usr/lib/libvfssl.a
/usr/x500/lib/libx509cms.a	/usr/lib/libx509cms.a
/usr/x500/lib/libpfx.a	/usr/lib/libpfx.a
/usr/x500/lib/libnspcommon.a	/usr/lib/libnspcommon.a
/usr/x500/lib/libcmskexp.a	/usr/lib/libcmskexp.a
/usr/x500/lib/libcms.a	/usr/lib/libcms.a
/usr/x500/lib/libvflicense.a	/usrlib/libvflicense.a
/usr/x500/lib/libcmskus.a	/usr/lib/libcmskus.a

Note: This document does not provide instructions on enabling two or more `commsrv` processes to act as front ends (with different characteristics) to the same DSA.

Chapter 4. Setting Up the Directory Schema

To set up the X.500 Directory schema (also referred to in places as subschema) to support Vault Registry, you need to modify the default schema for organizations, and add the necessary instances of organizations (CAs) and administrators to the Directory. To facilitate this process, templates can be used to do the following:

- Add CA instances at the organization level.
- Add CA password and privilege information.
- Create the Directory Administrator (`DirAdmin`) user password and privileges.

To set up the Directory schema, you must perform the following basic tasks:

1. Dump the existing schema to create the `org.tpl` template.
2. Edit the `org.tpl` schema template used to create the CA instances.
3. Using an editor, such as AIX's vi, create the CA password and privileges template, `add.caobjclass.pass.priv.tpl`.
4. Create the Directory Administrator (`DirAdmin`) password and privileges template, `add.DirAdmin.tpl`.

The sdua Format

The main command line tool for configuring the IBM eNetwork X.500 Directory and managing its data is the `sdua`. It is used here to dump existing schema, and from that schema, to build the instances of CAs and administrators required by Vault Registry. All `sdua` commands have a basic format that is used for both interactive and batch (file-based) commands. All commands end with a semicolon, and the attributes (such as `objectClass`) are separated with commas. Commas are also used within attribute values to separate the elements. The `sdua` format used within the schema is as follows:

```
action
       distinguished name
       Attribute {
               {List of attribute values}},
       Attribute {
               {List of attribute values}},
;
```

The following example shows the application of the `sdua` format to a real piece of a schema:

```
modify
      c "us" / o "ibm" / cn "subschema"
add values matchingRules {
    identifier { 2 5 13 34 },
    name { "certificateExactMatch" },
    information "certificateExactAssertion"
};
```

Note: More complicated attribute values are contained within braces as you will see in the "Creating and Editing the Templates" on page 13.

Dumping the Existing Organization Entry and Its Schema

This section assumes that your organization's X.500 Directory has been started using the default configuration, which contains a single country, Australia (au) and a single organization (deltawing). To build an organization template to be used later to create a new organization entry (in this case, a CA) with the appropriate schema, follow these steps:

1. In AIX, log in as the X.500 Directory Administrator (for example, x500usr) so that environment variables, including $VFHOME, are properly set.
2. Start the sdua tool by entering:

   sdua -gd
3. At the sdua prompt, enter the following command to display a list of countries:

   list ;

   The system returns a list of countries that looks like this (in this example the list contains only one item):

```
List Result {
     subordinates {
          {
              rdn {
                   countryName "au"
              }
          }
     }
}
```

4. To display a list of organizations within the target country (in this case au), enter the following command:

   list c"au" ;

   The system returns a list of countries that looks like this (in this example the list contains only one item):

```
List Result {
     subordinates {
          {
              rdn {
                   organizationName "deltawing"
              }
          }
     }
}
```

5. Exit sdua by entering **quit** at the sdua prompt.
6. Select one of the existing organization entries (in this case, deltawing) and create a temporary file (for example, /tmp/getOrg.tpl) containing the following commands:

   read c"au"/o"deltawing" return objectClass administrativeRole ;
   read c"au"/o"deltawing"/cn"subschema" return all ;
7. Obtain the organization entry and its subschema by entering:

   sdua -gd /tmp/getOrg.tpl > org.tpl

Creating and Editing the Templates

The most efficient approach to setting up the X.500 Directory schema is to create or edit template files as described in this section, copy the actual Vault Registry-specific text from this document, and insert it into the schema templates so that the Directory contains the information necessary to work with Vault Registry.

Editing the org.tpl Schema Template File

When you edit the org.tpl template file you created in “Dumping the Existing Organization Entry and Its Schema” on page 12 to be used to add Vault Registry-specific information, you will see the organization entry at the top of the file. The organization entry for the default eNetwork X.500 Directory looks like this:

```
entry
      countryName "au"
      / organizationName "deltawing"
with
      objectClass organization,
      administrativeRole autonomousArea
          subschemaAdminSpecificArea
          accessControlSpecificArea
          collectiveAttributeSpecificArea,
;
entry
      countryName "au"
      / organizationName "deltawing"
      / commonName "Subschema"
with
.
.
.
```

The schema is located beneath this section of the template.

To edit the `org.tpl` template file to modify the schema to add Vault Registry-specific information, follow these steps:

1. Add matching rules (`matchingRules`). See “Add matchingRules” on page 14.
2. Add attribute types (`attributeTypes`) and check for the existence of the comment and mail attribute types. See “Add and Modify attributeTypes” on page 14.
3. Add object classes (`objectClasses`). See “Add objectClasses” on page 16.
4. Add name forms (`nameForms`). See “Add nameForms” on page 17.
5. Edit the content rules (`dITContentRules`). See “Edit dITContentRules” on page 18.
6. Edit the structure rules (`dITStructureRules`). See “Add and Modify dITStructureRules” on page 20.

Notes:

1. Order within a section of the subschema does not matter except that if you add the entries at the end of a schema section, you must add a comma after the brace. The examples shown here assume that entries are not made at the end of a section.

2. Changes and additions are shown in bold.
3. It is recommended, for your convenience as well as for accuracy, that you copy the larger pieces of the template from this document and paste them into your schema.
4. Entries are case-sensitive. Exercise extreme care when editing the template file.

Add matchingRules

To add matching rules entries, scroll to the matchingRules section of the template illustrated below, and add the entries shown here in bold:

```
matchingRules {
               identifier { 2 5 13 5 },
               name {
                         "caseExactMatch"
               },
               information "DirectoryString {ub-match}"
       }
       {
              identifier { 2 5 13 34 },
              name {
                         "certificateExactMatch"
              },
              information "CertificateExactAssertion"
       }
       {
              identifier { 2 5 13 36 },
              name {
                         "certificatePairExactMatch"
              },
              information "CertificatePairExactAssertion"
       }
       {
              identifier { 2 5 13 38 },
              name {
                         "certificateListExactMatch"
              },
              information "CertificateListExactAssertion"
       }
```

Add and Modify attributeTypes

To add and modify attribute type entries, follow these steps:

1. Scroll to the attributeTypes section of the template as illustrated below, and add the attribute type entries shown here in bold:

```
attributeTypes {
                  identifier { 2 5 4 0 },
                  name {
                            "objectClass"
                  },
                  information {
                            equalityMatch objectIdentifierMatch,
                            attributeSyntax "OBJECT IDENTIFIER"
                  }
            }
            {
                  identifier { 2 5 4 36 },
                  name {
                            "userCertificate"
                  },
                  information {
                            equalityMatch certificateExactMatch,
                            attributeSyntax "Certificate"
                  }
```

```
}
{
        identifier { 2 5 4 37 },
        name {
                "cACertificate"
        },
        information {
                equalityMatch certificateExactMatch,
                attributeSyntax "Certificate"
        }
 }
 {
        identifier { 2 5 4 38 },
        name {
                "authorityRevocationList"
        },
        information {
                equalityMatch certificateListExactMatch,
                attributeSyntax "CertificateList"
        }
 }
 {
        identifier { 2 5 4 39 },
        name {
                "certificateRevocationList"
        },
        information {
                equalityMatch certificateListExactMatch,
                attributeSyntax "CertificateList"
        }
 }
 {
        identifier { 2 5 4 40 },
        name {
                "crossCertificatePair"
        },
        information {
                equalityMatch certificatePairExactMatch,
                attributeSyntax "CertificatePair"
        }
  }
  {
        identifier { 2 5 4 53 },
        name {
                "deltaRevocationList"
        },
        information {
                equalityMatch certificateListExactMatch,
                attributeSyntax "CertificateList"
        }
  }
  {
        identifier { 1 2 840 113533 7 68 10 },
        name {
                "attributeCertificate"
        },
        information {
                attributeSyntax "AttributeCertificate"
        }
   }
```

2. Check for the existence of the comment attribute type. If it is not present, add the following entry under attributeTypes:

```
   {
      identifier { 1 3 32 0 2 0 4 46 },
      name {
                "comment"
```

```
                    },
                    information {
                              equalityMatch caseExactMatch,
                              substringsMatch caseExactSubstringsMatch,
                              attributeSyntax "DirectoryString"
                    }
                }
```

3. Check for the existence of the mail or email attribute type. If it is not present, add the following entry under attributeTypes. If it is present, but the emailAddress attribute type is not first, edit the entry so that emailAddress comes first:

```
{
                  identifier { 1 3 32 0 2 0 4 52 },
                      name {
                              "emailAddress",
                              "mail",
                              "email"
                      },
                      information {
                              derivation name,
                              attributeSyntax "DirectoryString"
                      }
              }
```

Add objectClasses

To add object class entries, scroll to the objectClasses section of the template illustrated below, and add the entries shown here in bold:

```
objectClasses {
                identifier { 2 5 6 0 },
                name {
                          "top"
                },
                information {
                          kind abstract,
                          mandatories {
                                  objectClass
                          }
                }
        }
        {
                identifier { 1 2 840 113533 7 67 0 },
                name {
                          "entrustUser"
                },
                information {
                          subclassOf {
                                  top
                          },
                          kind auxiliary,
                          optionals {
                                  userCertificate
                          }
                }
      }
      {
                identifier { 1 2 840 113533 7 67 1 },
                name {
                          "entrustCA"
                },
                information {
                          subclassOf {
                                  top
                          },
                          kind auxiliary,
```

```
                        optionals {
                                cACertificate,
                                certificateRevocationList,
                                authorityRevocationList,
                                crossCertificatePair,
                                attributeCertificate
                        }
                }
        }
        {
                identifier { 2 5 6 19 },
                name {
                        "cRLDistributionPoint"
                },
                information {
                        subclassOf {
                                top
                        },
                        mandatories {
                                commonName
                        },
                        optionals {
                                certificateRevocationList,
                                authorityRevocationList,
                                deltaRevocationList
                        }
                }
        }
        {
                identifier { 1 2 840 113533 7 67 4 },
                name {
                        "uniquelyIdentifiedUser"
                },
                information {
                        subclassOf {
                                top
                        },
                        kind auxiliary,
                        mandatories {
                                serialNumber
                        }
                }
        }
```

Add nameForms

To add name form entries, scroll to the nameForms section of the template illustrated below, and add the entries shown here in bold:

```
nameForms {
            identifier { 2 5 15 3 },
            name {
                    "orgNameForm"
            },
            information {
                    subordinate organization,
                    namingMandatories {
                            organizationName
                    }
            }
      }
      {
            identifier { 2 5 15 14 },
            name {
                    "cRLDistributionPointNameForm"
            },
            information {
```

```
                subordinate cRLDistributionPoint,
                namingMandatories {
                        commonName
                }
        }
}
{
        identifier { 1 3 32 0 2 100 15 0 },
        name {
                "orgPersonNF2"
        },
        information {
                subordinate organizationalPerson,
                namingMandatories {
                        commonName,
                        serialNumber
                }
        }
}
{
        identifier {1 3 32 0 2 100 1000 1000},
        name {
                "orgPersonNF3"
        },
        information {
                subordinate organizationalPerson,
                namingMandatories {
                        commonName,
                        emailAddress
                }
        }
}
{
        identifier {1 3 32 0 2 100 1000 1001},
        name {
                "orgPersonNF4"
        },
        information {
                subordinate organizationalPerson,
                namingMandatories {
                        commonName,
                        serialNumber,
                        emailAddress
                }
        }
}
```

Edit dITContentRules

To edit content rules entries, scroll to the dITContentRules section of the template illustrated below, and add the entries shown here in bold:

```
dITContentRules {
        structuralObjectClass organizationalPerson,
        auxiliaries {
                entrustCA,
                entrustUser,
                uniquelyIdentifiedUser
        },
        optional {
                businessCategory,
                givenName,
                location,
                mailingAddress,
                facsimileNumber,
                comment,
                keylinkAddress,
```

```
                    emailAddress,
                    mhs-or-addresses,
                    employeeNumber,
                    mobileNumber,
                    pagerNumber,
                    billingCode,
                    employerCode,
                    printCode,
                    sortSubs,
                    telecomTelexNumber,
                    serialNumber
          },
          name {
                    "deltawingOrganizationalPerson"
          }

}
{
          structuralObjectClass organizationalRole,
          auxiliaries {
                entrustCA,
                entrustUser,
                uniquelyIdentifiedUser
          },
          optional {
                    businessCategory,
                    location,
                    mailingAddress,
                    facsimileNumber,
                    comment,
                    emailAddress,
                    mhs-or-addresses,
                    mobileNumber,
                    pagerNumber,
                    billingCode,
                    employerCode,
                    printCode,
                    sortSubs,
                    telecomTelexNumber
          },
          name {
                    "deltawingOrganizationalRole"
          }
}
{
          structuralObjectClass organization,
          auxiliaries {
                entrustCA
        },
          optional {
                    manager,
                    businessCategory,
                    location,
                    mailingAddress,
                    facsimileNumber,
                    comment,
                    emailAddress,
                    mhs-or-addresses,
                    billingCode,
                    employerCode,
                    printCode,
                    sortSubs,
                    telecomTelexNumber
          },
          name {
                    "deltawingOrganization"
          }
```

```
        }
{
                structuralObjectClass organizationalUnit,
                auxiliaries {
                        entrustCA
                },

                optional {
                        manager,
                        location,
                        mailingAddress,
                        facsimileNumber,
                        comment,
                        billingCode,
                        employerCode,
                        emailAddress,
                        printCode,
                        sortSubs,
                        telecomTelexNumber
                },
                name {
                        "deltawingOrganizationalUnit"
                }
        }
```

Add and Modify dITStructureRules

To add and modify content rule entries, follow these steps:

1. Complete the following worksheet by adding your rule numbers to the last column.

   This worksheet documents the rule identifiers for existing DIT structure rules and the new rule identifiers you will be adding. Enter a rule number for each corresponding rule name that already exists in the dITStructureRules section of the template. Then, enter a rule number for each corresponding rule name for the rules that you will be adding. All rule identifiers must be unique so that any currently unused value is acceptable for the new rules.

Table 2. Structure Rules

Rule Name	Default Rule Identifer	Your Rule Identifier
Existing Rules		
orgNameForm	1	
orgUnitNameForm	2	
orgPersonNameForm	3	
orgRoleNameForm	9	
Rules To Be Added		
cRLDistributionPointNameForm	99	
orgPersonNF2	100	
orgPersonNF3	10020	
orgPersonNF4	10030	

2. Scroll to the dITStructureRules section of the template illustrated, where you should see an initial rule that looks like this:

```
dITStructureRules {
                ruleIdentifier 1,
                nameForm orgNameForm,
```

```
                name {
                       "deltawingOrganizationStructureRule"
                }
        }
```

3. Replace the bold numbers shown here in the dITStructureRules section of the template with the corresponding numbers you added to the worksheet in Table 2 on page 20:

```
        {
                ruleIdentifier 99,
                nameForm cRLDistributionPointNameForm,
                superiorStructureRules {
                        1,
                        2,
                        3,
                        100,
                        10020,
                        10030,
                        9
                },
                name {
                        "cRLDistributionPointStructureRule"
                }
        }
        {
                ruleIdentifier 100,
                nameForm orgPersonNF2,
                superiorStructureRules {
                        1,
                        2
                },
                name {
                       "OrganizationalPersonStructureRule2"
                }
        }
        {
                ruleIdentifier 10020,
                nameForm orgPersonNF3,
                superiorStructureRules {
                        1,
                        2
                },
                name {
                        "OrganizationalPersonStructureRule3"
                }
        }
        {
                ruleIdentifier 10030,
                nameForm orgPersonNF4,
                superiorStructureRules {
                        1,
                        2
                },
                name {
                        "OrganizationalPersonStructureRule4"
                }
        }
```

Creating the add.caobjclass.pass.priv.tpl Template File

To create a template file for adding the CA object class (entrustCA), password, and privilege to the X.500 Directory entries, using an editor such as AIX vi, enter the text shown in this example into a file named add.caobjclass.pass.priv.tpl.

Note: Entries are case-sensitive. Exercise extreme care when creating the template file.

```
modify
        c"US"/o"IBM"
add values objectClass entrustCA
;
modify
        countryName"US"
        / organizationName"IBM"
remove privilege
;
modify
        countryName"US"
        / organizationName"IBM"
remove userPassword
;
modify
        countryName"US"
        / organizationName"IBM"
add userPassword
               "Secure98"
add privilege {
              accessLevel
                        update
              ,
              subtree {
                        countryName"US"
                        / organizationName"IBM"
              }
}
;
```

Creating the addDirAdmin.tpl Template File

To create a template file for adding the Directory Administrator entry and its associated password and privilege to the X.500 Directory, using an editor such as AIX vi, enter the text shown in this example into a file named `add.DirAdmin.tpl`.

Note: Entries are case-sensitive. Exercise extreme care when creating the template file.

```
entry
     countryName"US"
     / organizationName"IBM"
     / commonName"DirAdmin"
with
     objectClass organizationalPerson
           person,
           userPassword
                  "Secure98",
     privilege {
               accessLevel admin,
               subtree {
                    countryName"US"
                    / organizationName"IBM"
               }
}
;
```

Chapter 5. Configuring the Directory

To configure the Directory, you must complete the following basic tasks:

1. Complete the worksheet for the Directory by adding the values your organization plans to use that correspond to the Vault Registry default values. See "Completing the Worksheet for Directory Entries".
2. Log in as the X.500 Directory Administrator (for example, `x500usr`) so that environment variables are properly set.
3. Add the country in which your organization is located. See "Adding the Country Entry" on page 24 for instructions.
4. Add a Vault CA at the organization level, and for ease of use, edit the following three files to reflect the values for that CA:
   - `org.tpl`. See "Adding the CA Entries at the Organization Level" on page 25 for instructions.
   - `add.caobjclass.pass.priv.tpl`. See "Adding the CA Object Class, Password, and Privilege for Each CA" on page 26 for instructions.
   - `add.DirAdmin.tpl`. See "Adding the Directory Administrator Entry, Password, and Privilege for Each CA" on page 27 for instructions.
5. Add the first Organization CA you plan to use, and edit the files listed in the previous step to reflect the values for that CA.
6. Add any additional Organization CAs, and for each CA, edit the files listed in step 4 to reflect the values for that CA.

Completing the Worksheet for Directory Entries

This worksheet is designed to facilitate configuration of the Directory. Complete the worksheet for your Directory by adding the values your organization plans to use that correspond to the Vault Registry and eNetwork X.500 Directory default values. Add these values to the last column of Table 3. Then, use the values that you specify to replace the existing values in the template files as described in the instructions that follow.

Note: Entries are case-sensitive.

Table 3. Worksheet for Directory Entries

Entry Name	**Default Vault Registry and eNetwork X.500 Directory Values**	**Your Organization's Value**
Directory user name	x500usr	
Country name (countryName)	US au	
Vault CA (vaultCA) Values:		
Organization name (organizationName)	IBM deltawing	

Table 3. Worksheet for Directory Entries (continued)

Entry Name	Default Vault Registry and eNetwork X.500 Directory Values	Your Organization's Value
CA password	Secure98	
Directory Administrator common name (commonName)	DirAdmin	
Directory Administrator password	Secure98	
First Organization CA (orgCA) Values:		
organizationName	IBM1	
CA password	Secure98	
Directory Administrator commonName	DirAdmin	
Directory Administrator password	Secure98	
Template File Names		
CA (organization level) schema	org.tpl	
CA password and privileges schema	add.ca.pass.priv.tpl	
Directory Administrator password and privileges schema	add.DirAdmin.tpl	

Adding the Country Entry

To add a new country entry to your Directory, follow these steps:

1. In AIX, enter the following commands to display a list of existing countries:

   ```
   sdua -gd
   ```

2. At the sdua prompt, enter:

   ```
   list ;
   ```

   The system returns a list of countries that looks like this:

   ```
   List Result {
        subordinates {
             {
                  rdn {
                      countryName "au"
                  }
             }
        }
   }
   ```

3. Exit sdua by entering **quit** at the sdua prompt.
4. Select one of the existing country entries (for example, au) and create a temporary file (for example, /tmp/newCountry1) containing the following commands:

   ```
   read c"au" return objectClass administrativeRole ;
   read c"au"/cn"subschema" return all ;
   ```

5. Obtain the sample template country entry (in this case, newCountry2) and its subschema by entering:

   ```
   sdua -gd /tmp/newCountry1 > /tmp/newCountry2
   ```

6. Change c"au" to the new country abbreviation (c*"new country abbreviation"* such as c"US" for the United States) by editing the newCountry2 file:

```
vi /tmp/newCountry2
```

Changes occur in two places. Make sure that US is capitalized.

7. Load the new country entry as follows:

   ```
   sdua -gd /tmp/newCountry2
   ```

   If successful, this command returns two NULL results.

8. To verify that the entry has been made, display the list of existing countries and check that the new entry, in this case US, has been added:

   ```
   sdua -gd
   ```

9. At the sdua prompt, enter:

   ```
   list ;
   ```

Adding the CA Entries at the Organization Level

To add the CA entries at the organizational level, perform the following steps. In doing so, pay close attention to the case of your entries.

1. Add the Vault CA, as follows:
   a. Edit the org.tpl file to reflect your country and your organization's name for the Vault CA. The default Vault Registry name for the Vault CA is IBM.
   b. Make sure the changes you made were correct by entering the following command at the AIX command line:

      ```
      sdua -gU org.tpl
      ```

      This command checks the format. If there are no errors, the command returns the AIX prompt.
   c. To apply your changes to the Directory, enter this command:

      ```
      sdua -gd org.tpl
      ```

      If successful, this command returns two NULL results.
2. Add the first Organization CA you plan to have, as follows:
   a. Edit the org.tpl file to reflect your organization's name for the first Organization CA. The default Vault Registry name for the first Organization CA is IBM1.
   b. Make sure the changes you made were correct by entering the following command at the AIX command line:

      ```
      sdua -gU org.tpl
      ```

      This command checks the format. If there are no errors, the command returns the AIX prompt.
   c. To apply your changes to the Directory, enter this command:

      ```
      sdua -gd org.tpl
      ```

      If successful, this command returns two NULL results.
3. Repeat this procedure for each additional Organization CA used in your configuration.
4. Edit the $VFHOME/setup/config file.
5. Set the base entry which, for US as the country and Vault CA name as the organization, would look like this:

   ```
   baseentry = { /c"US"o"IBM"}
   ```

Note: The DN for the schema itself would look like /c"US"/o"CAname"/cn"subschema". This entry belongs to the objectClass subschema.

Edit the $VFHOME/setup/sdua.startup file.

For a country named US and an organization named CAname, enable sdua scripts to use the base schema obtained from the base object set in the startup file by entering:

```
set base object to /c"US"/o"CAname"
```

Adding the CA Object Class, Password, and Privilege for Each CA

To add the CA object class, password, and privilege, perform the following steps. In doing so, pay close attention to the case of your entries.

1. Add the Vault CA object class (entrustCA), password, and privilege, as follows:
   a. Edit the add.caobjclass.pass.priv.tpl file to reflect your country and your organization's name and password for the Vault CA. The default Vault Registry name for the Vault CA is IBM and the password is Secure98. Changing the country and organization names within the subtree definition in the template sets up the proper privilege for the CA.
   b. Make sure the changes you made were correct by entering the following command at the AIX command line:

      ```
      sdua -gU add.caobjclass.pass.priv.tpl
      ```

      This command checks the format. If there are no errors, the command returns the AIX prompt.
   c. To apply your changes to the Directory, enter this command:

      ```
      sdua -gd add.caobjclass.pass.priv.tpl
      ```

      If successful, this command returns four results. The first is NULL, the second and third are expected errors, and the fourth is another NULL.
2. Add the first Organization CA object class, password, and privilege, as follows:
   a. Edit the add.caobjclass.pass.priv.tpl file to reflect your country and the name and password of the first Organization CA. The default Vault Registry name for the first Organization CA is IBM1 and the password is Secure98. Changing the country and organization names within the subtree definition in the template sets up the proper privilege for the CA.
   b. Make sure the changes you made were correct by entering the following command at the AIX command line:

      ```
      sdua -gU add.caobjclass.pass.priv.tpl
      ```

      This command checks the format. If there are no errors, the command returns the AIX prompt.
   c. To apply your changes to the Directory, enter this command:

      ```
      sdua -gd add.caobjclass.pass.priv.tpl
      ```

      If successful, this command returns four results. The first is NULL, the second and third are expected errors, and the fourth is another NULL.
3. Repeat this procedure, with appropriate passwords and privileges, for each additional Organization CA used in your configuration .

Adding the Directory Administrator Entry, Password, and Privilege for Each CA

To add the Directory Administrator entry, password, and privilege, perform the following steps. In doing so, pay close attention to the case of your entries.

1. Add the Vault CA Directory Administrator entry, password, and privilege, as follows:
   a. Edit the `add.DirAdmin.tpl` file to reflect your country, your organization's name, and the name and password for the Vault CA Directory Administrator. The default Vault Registry name for the Vault CA is `IBM`, the Directory Administrator name is `DirAdmin`, and the Directory Administrator password is `Secure98`. Changing the country and organization names within the subtree definition in the template sets up the proper privilege for the Directory Administrator.
   b. Make sure the changes you made were correct by entering the following command at the AIX command line:

      ```
      sdua -gU add.DirAdmin.tpl
      ```

      This command checks the format. If there are no errors, the command returns the AIX prompt.
   c. To apply your changes to the Directory, enter this command:

      ```
      sdua -gd add.DirAdmin.tpl
      ```

      If successful, this command returns one NULL result.
2. Add the first Organization CA Directory Administrator entry, password, and privilege, as follows:
   a. Edit the `add.DirAdmin.tpl` file to reflect your country and the name and password of the first Organization CA. The default Vault Registry name for the first Organization CA is `IBM1` and the password is `Secure98`. Changing the country and organization names within the subtree definition in the template sets up the proper privilege for the Directory Administrator.
   b. Make sure the changes you made were correct by entering the following command at the AIX command line:

      ```
      sdua -gU add.DirAdmin.tpl
      ```

      This command checks the format. If there are no errors, the command returns the AIX prompt.
   c. To apply your changes to the Directory, enter this command:

      ```
      sdua -gd add.DirAdmin.tpl
      ```

      If successful, this command returns one NULL result.
3. Repeat this procedure for each additional Organization CA used in your configuration with the appropriate Directory Administrator's DN, password, and privilege.

Chapter 6. Monitoring the X.500 Directory

Vault Registry is now a Directory-neutral product. Manual monitoring is unnecessary because the Vault Registry Monitor program attempts connection to the Directory often enough to indicate whether the Directory is unavailable for any reason. For information on troubleshooting, specifically checking logging data for query, update, and connection transactions, refer to the documentation appropriate for the Directory used by your organization.

Chapter 7. Backing Up and Restoring the X.500 Directory

You should be aware of the options for backing up the Directory, either in its entirety, or by subtrees. This chapter provides suggestions for backing up and restoring the X.500 Directory that you can explore further by referring to your site's X.500 Directory documentation. Due to the Directory-neutral nature of Vault Registry, discussion of the complications associated with the backing up of distributed and replicated Directories is outside the scope of this document.

Backup Suggestions

Following are suggestions for performing Directory backups:

- You can use the `sdua dump` command as an online command to back up a subtree or the entire contents of the Directory in `sdua` format.
- You can use the `save` command as an online command to take a snapshot of the whole Directory database. In this case, data is saved in neither `sdua` nor LDAP Directory Interchange Format (LDIF). Be sure to verify that the `data/__safe` file is backed up as well. If it is not, copy it to the backup storage area.
- Your site can develop an `ldapsearch`-based utility that sorts the data into a tree structure so the data (in LDIF) can be loaded via `ldapadd`. The Vault Registry migration tool uses this approach.

Restore Suggestions

Following are suggestions for restoring the Directory database:

- If the backed up data is in `sdua` format and you want to perform an online restore, use the `sdua` command.
- If the backed up data is in `sdua` format and you want to perform an offline fast load restore, you can use the `vfload` command.
- If the data was backed up using the `save` command, copy back the data Directory and the `__safe` file to restore the Directory.
- If the backed up data is in LDIF, load the data via the `ldapadd` API or utilities to restore the Directory.

Program Number: 5648–B28 - Document Last Updated May 25, 1999.

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.