ViewDS Installation and Operation Guide

Published: 2015
Version: 7.4(3)

ViewDS Installation and Operation Guide

September 2015

Document Lifetime
ViewDS may occasionally update online documentation between software releases. Consequently, this PDF may not contain the most up-to-date information. Refer to the online documentation at www.viewds.com/resources/documentation.html for the most current information.

About this guide

This guide includes an overview of ViewDS, instructions for installing and configuring ViewDS, and instructions for day-to-day operational tasks. It also includes 'key concepts' chapters that provide the background theory required to work with different aspects of ViewDS.

This section includes:

- Who should read this guide
- Related documents
- How this guide is organized

Who should read this guide

Read this guide if you are responsible for installing or administrating ViewDS.

It provides the following:

- an overview of ViewDS, its components and how they work together
- how to install and configure ViewDS
- how to install and explore an example directory
- an overview of the day-to-day ViewDS tasks including how to start, stop, and monitor ViewDS operations
- an overview of ViewDS concepts
- an overview of adapting ViewDS to your requirements

> You will need some familiarity with the host operating system (Solaris, Linux or Windows).

Related documents

The other documents in the ViewDS document set are:

- ViewDS Access Proxy Installation Guide
- ViewDS Access Sentinel Installation and Reference Guide
- ViewDS Application Integration Kit for .NET
- ViewDS Application Integration Kit for Java
- ViewDS Technical Reference Guide: Directory System Agent

- ViewDS Technical Reference Guide: User Interfaces
- ViewDS Management Agent In-application Help

How this guide is organized

This guide contains the following:

Section 1: About this guide

Provides an overview of this guide.

Section 2: System overview

Provides an overview of the system and an introduction to the processes, utilities and files in a ViewDS installation.

Section 3: Installing ViewDS

Provides instructions for installing ViewDS.

Section 4: Configuring ViewDS

Provides instructions for configuring ViewDS.

Section 5: Exploring the example directory

Provides instructions for exploring the example directory supplied with ViewDS.

Section 6: Operating the directory

Provides instructions for day-to-day operations including how to start and stop ViewDS, and how to backup and restore.

Section 7: Key concepts – Schema

Introduces the concepts required to work with ViewDS schema, how to manage schema through the ViewDS Management Agent, and includes high-level guidance to help you adapt ViewDS to your requirements.

Section 8: Key concepts – Security

Introduces the concepts required to work with ViewDS security, how to manage security through the ViewDS Management Agent, and includes high-level guidance to help you adapt ViewDS to your requirements.

Section 9: Key concepts – Distribution and replication

Introduces the concepts required to work with ViewDS distribution and replication, and how to manage distribution and replication through the ViewDS Management Agent

System overview

This section provides an overview of the major features and components of ViewDS. You will need this background information before installing and configuring ViewDS.

Topics covered in this section include:

- What is ViewDS
- ViewDS components

What is ViewDS

ViewDS is a standards-based directory service. It delivers superior capabilities in both traditional (white pages) and emerging (XML and B2B) technologies, and provides high-performance complex searching on object-oriented hierarchical data.

By complying with X.500, LDAP and XML Enabled Directory (XED) standards, ViewDS provides a rich set of features offering many benefits for directory users and administrators. The major features of ViewDS are outlined in the following topics:

- Improved user support
- Data represents your real-world hierarchy
- XACML integration
- Flexible data search
- PKI matching rules
- Windows-based management tool
- Certificate look-up service
- Reliability with scalability
- Integrates with your existing technology

Improved user support

ViewDS makes users' access to a directory easier by providing a web-based client and by supporting advanced approximate matching.

Web-based client

The ViewDS web-based client, Access Presence, reduces the time required for implementation and support by providing client functionality that can be customized
to your requirements.

This functionality includes:

- Organization charts
- Self-service portals
- Reporting interfaces
- Certificate management
- Chinese language approximate matching

Approximate matching

Users are often imprecise when they search a directory – they might, for example, misspell or mistype names, or use acronyms or abbreviations. ViewDS delivers superior approximate matching that supports a range of strategies and provides better service to users:

- phonetic matching – for example, the search criteria ‘pane’ would match ‘payne’
- typing correction – ‘Dircetor’ would match 'Director'
- stem matching – ‘optics’ would match ‘optical’
- synonym matching – 'Bob' would match 'Robert', and 'road' would match 'street'
- abbreviation matching – 'NSW' would match 'New South Wales'

Additionally, specialized indexes are used to rapidly evaluate approximate matches.

Data represents your real-world hierarchy

ViewDS minimizes the time, resources and planning required to make large-scale changes to a Directory Information Tree (DIT).

Directory Information Tree

Directory entries are arranged in a hierarchy called the DIT. A directory is of most use when its DIT mirrors a hierarchy in the real-world, such as the structure of a company, health trust or government department.

Real world hierarchies, however, tend to change very frequently – for example, departments can be formed, dissolved, moved, merged or split. This can lead directory administrators to flattening and simplifying the DIT to avoid the complexity of maintaining it. Unfortunately, the result can be little more than a list of entries, rather than a resource that helps people understand the structure of their organization.

ViewDS overcomes this by providing seamless support for changes to complex hierarchies. Directory administrators can change a DIT simply and rapidly so that it always accurately reflects the real-world structure of their organization.

Move and rename

Unlike many directories, ViewDS allows DIT entries with subordinates (non-leaf entries) to be moved or renamed, while looking after all the associated links such as 'managed by' and 'manager of' relationships. ViewDS can be configured to enforce various referential integrity constraints.

XACML integration

ViewDS leverages its XML capabilities to allow XACML policy to be defined that conforms to XACML Version 3.0. This standard describes a language for expressing access controls and a protocol for requesting access-control decisions, both written in XML.

ViewDS includes the XACML Access Control scheme, which allows XACML policy to be applied to the ViewDS directory.

ViewDS Access Sentinel extends the capability to allow XACML policy to be applied to applications external to ViewDS permitting unified access control across disparate applications and data sources, improving the user experience and reducing administration and support. For more information about Access Sentinel see the ViewDS Access Sentinel Installation and Reference Guide.

Flexible data search

The ViewDS component-matching rules allow users to search specific parts of an attribute with a complex syntax – for example, digital certificates, XML documents and certificate-revocation lists.

To illustrate, consider a Human Resources department that stores employees' resumes as XML documents. Component matching would allow a specific area of
the resumes to be searched individually, and therefore, efficiently. For example, a user might search on just the 'qualifications' component of the resumes to find employees with a qualification in nuclear physics.

Without component matching, an application developer would generally need to either scan an entire directory or filter search results in order to find the required data. Both options are inefficient and slow.

PKI matching rules

X.509 certificates can be stored by most directories, but few support matching rules for the X.509 Public Key Infrastructure (PKI) attribute syntaxes. Consequently, environments with large certificate-revocation lists can suffer unacceptable performance.

ViewDS supports the PKI matching rules, and also supports component matching for the PKI syntaxes. A PKI application that uses ViewDS will process certificates faster and with less effort. ViewDS also

supports strong authentication with X.509 certificates for client-to-server and server-to-server authentication.

Windows-based management tool

The ViewDS Management Agent is a Windows-based application that makes managing Directory System Agents (DSAs) simpler and more efficient.

Certificate lookup service

ViewDS Access Proxy provides a certificate lookup service that complies with TSCP specifications. For more information about Access Proxy, see the ViewDS Access Proxy Installation Guide.

Reliability with scalability

ViewDS is designed to accommodate 'mission-critical' directory applications and is designed for continuous operation. During routine maintenance – such as backing up the database and check-pointing update logs – ViewDS continues to process queries.

ViewDS is designed to be scalable:

- A single ViewDS server can accommodate many millions of entries.
- There are no restrictions on the number of entries, size or number of attributes, depth of the DIT or number of connected users (other than those imposed by the host's operating system and hardware).
- ViewDS has an optimized tool for fast loading of entries. During bulk loading, the difference between the load-speed for the first and last entries loaded is negligible.
- ViewDS restarts rapidly after a power failure, irrespective of the number of entries.

Integrates with your existing technology

The ViewDS standards-based architecture ensures seamless communication with third-party directory clients and servers:

- Implements OSI Stack (including RFC1006 over TCP/IP) and X.500 Internet Directly Mapped Protocol (IDMP).
- Supports LDAP, XLDAP, IDMP, XIDMP, SPML, SNMP and HTTP access.

Additionally, ViewDS components communicate using an efficient proprietary protocol running over TCP/IP.

ViewDS components

Figure 1: ViewDS components shows the main components of ViewDS.

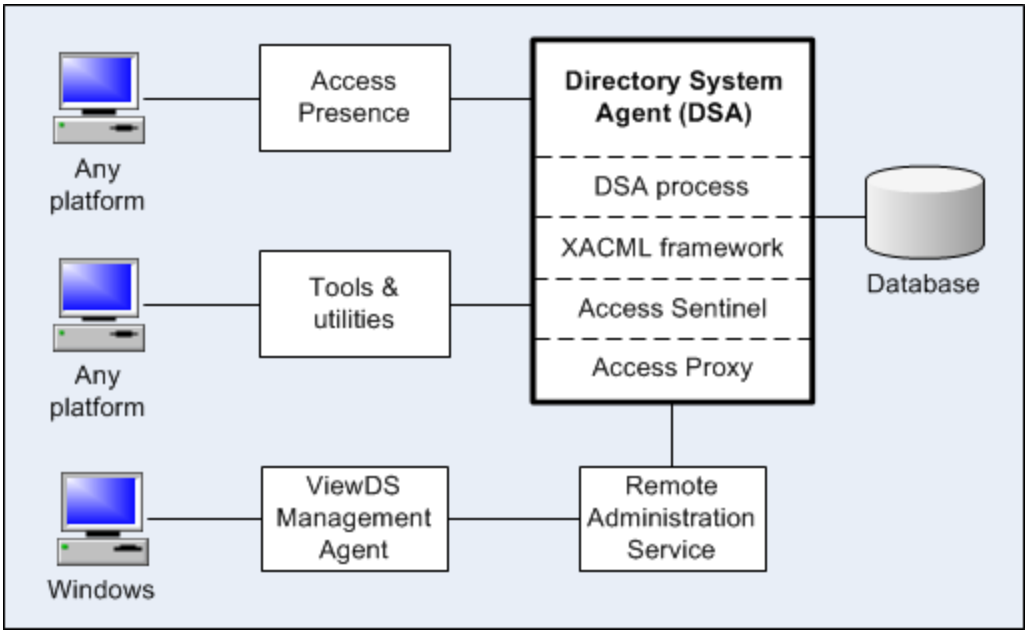


Figure 1: ViewDS components

The following topics provide an overview of these components:

- DSA process
- XACML framework
- Access Sentinel
- Access Proxy
- Access Presence
- Tools and utilities
- ViewDS Management Agent
- Remote Administration Service
- SNMP Proxy Agent

DSA process

The DSA process is a multi-threaded process comprising the main dsa thread and one or more dot threads.

dsa

Handles all communications with client applications and peer servers. It initiates and controls the state of the dot threads, queues and coordinates the processing of requests, and can manage the state of the database.

dot

The dot (Directory Operation Thread) threads receive and process requests from the dsa. The dot threads will retrieve or update data from the database in order to satisfy requests from client applications. They also provide ViewDS's flexible searching capabilities.

The number of dot threads running can be configured; the default number of dot threads is three and the maximum is 5.

XACML framework

The XACML framework conforms to the XACML Version 3.0 standard and facilitates the XACML Access Control scheme. This scheme allows you to define XACML policy that imposes fine-grained access control on the directory.

The framework has four logical components:

- Policy Enforcement Point (PEP) – protects directory entries from unauthorized actions (view, modify, etc.).
- Policy Decision Point (PDP) – makes decisions about whether access should be granted to a directory entry.
- Policy Administration Point (PAP) – allows XACML policy to be managed and stored. The ViewDS Management Agent allows you to create and manage policy, which are stored in the directory.
- Policy Information Policy Point (PIP) – stores additional information, such as user attributes, which the PDP can use to make access-control decisions. This information is stored in the directory.

Figure 3: ViewDS XACML framework shows the logical components:

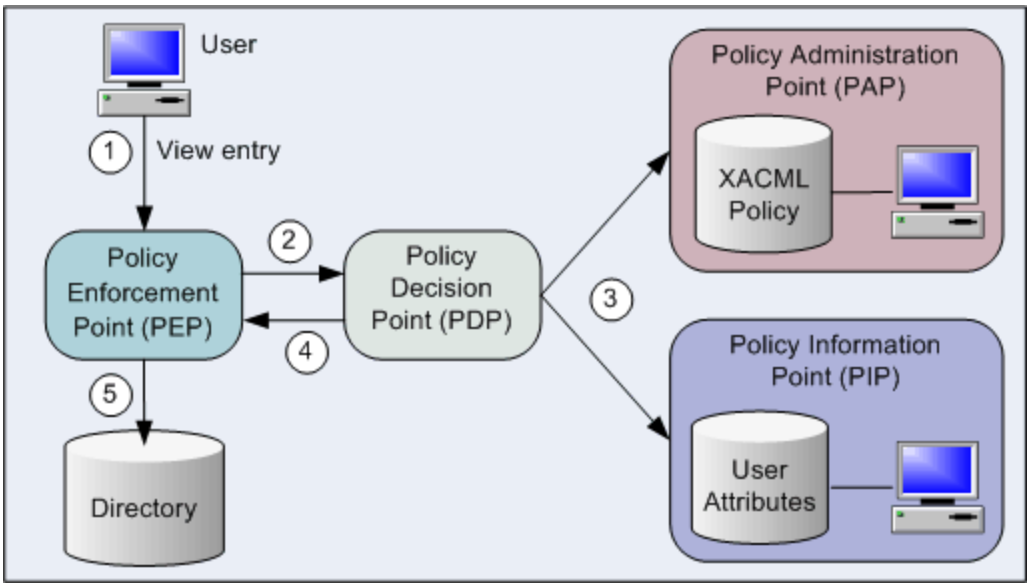


Figure 3: ViewDS XACML framework

The PDP and PEP are part of the DSA. The three data sets shown above are all stored in the same ViewDS database, and can be managed through the same user interface, the ViewDS Management Agent.

The steps in Figure 3: ViewDS XACML framework are as follows:

1. A user attempts to view an entry in the ViewDS directory.
2. The PEP sends an 'authorization decision request' to the PDP, which includes information such as the user's identity and entry they are attempting to view.
3. The PDP determines whether access should be permitted. It looks at the data in the request, the appropriate XACML policy, and user attributes in the PIP.
4. An 'authorization decision response' is returned to the PEP, which then acts on the decision to permit or deny access.

For information about writing XACML policy, see Key concepts – Security.

Access Sentinel

ViewDS Access Sentinel extends the XACML framework to allow you to apply XACML policy to applications external to ViewDS. It also includes an additional PAP interface, the Authorization Policy Manager, which allows XACML policy to be managed from any platform.

For more information about Access Sentinel, see the ViewDS Access Sentinel Installation and Reference Guide.

Access Sentinel requires additional licensing.

Access Proxy

ViewDS Access Proxy provides a certificate lookup service that complies with TSCP specifications. For more information about Access Proxy, see the ViewDS Access Proxy Installation Guide.

Access Proxy requires additional licensing.

Access Presence

Access Presence (webdua.cgi) operates with a standard web server and allows users to access a ViewDS directory through their web browser.

You can configure many aspects of Access Presence on the client and server side. The ViewDS Management Agent provides access to the server-side configuration, including specification of how entries and their attributes are displayed.

Access Presence requires additional licensing.

Tools and utilities

ViewDS includes the following command-line tools and utilities:

- Stream DUA
- DSA Controller
- Printing DUA
- ViewDS tools

Stream DUA

The Stream DUA is a tool for managing a directory when the ViewDS Management Agent is unavailable. It is a batch-oriented Directory User Agent (DUA) for accessing directory data and managing backups, schema and knowledge. It is a text-oriented DUA that accepts input commands and data in a stream form and can therefore run non-interactively.

DSA Controller

The DSA Controller is a single process, the dsac process, and an important tool for controlling a DSA when the ViewDS Management Agent is unavailable. It runs on the same host as the DSA, and allows you to query and modify the DSA's status. This tool should only be made available to the ViewDS system administrator.

Printing DUA

The Printing DUA extracts data from ViewDS and prepares it for other applications. It can sort data, tag data items, insert text and other formatting information. The resulting data can then be used, for example, with a desktop publishing package to produce a printed directory listing.

ViewDS tools

As well as the ViewDS Management Agent, there are several command-line tools for backup, communications, and data loading – some are shell scripts, others are stand-alone programs:

- `dbbackup` – this script performs an incremental or full backup of a directory to a tape device (Solaris and Linux only).
- `smerge` – this script sorts and merges the entries in the update log files so that a database can be restored from backup files without losing recent updates.
- `vfload` – this is the ViewDS Fast Load utility, which allows dump files to be loaded into the directory.

The utilities are described fully in the ViewDS Technical Reference Guide: Directory System Agent.

ViewDS Management Agent

The ViewDS Management Agent is a Windows-based application that runs on a different computer to the DSA's host. It allows you to manage the status of one or more DSAs remotely and access their configuration parameters, log files, directory data, schema, knowledge and access controls.

To ensure secure administration, ViewDS uses SSL/TLS connections and certificate-based authentication between the ViewDS Management Agent and DSA, and between the ViewDS Management Agent and Remote Administration Service (RAS).

Remote Administration Service

The Remote Administration Service (RAS) runs on the DSA's host computer. It allows the ViewDS Management Agent, installed on a remote computer, to start and stop the DSA, modify its configuration and

view its log files.

SNMP Proxy Agent

The SNMP proxy agent is a tool designed to collect the MIB objects from a number of DSAs and combine them to present a single MIB object to an SNMP management console. This has the effect of having a distributed DSA environment appear to be a single DSA to the SNMP manager.

What is ViewDS

ViewDS is a standards-based directory service. It delivers superior capabilities in both traditional (white pages) and emerging (XML and B2B) technologies, and provides high-performance complex searching on object-oriented hierarchical data.

By complying with X.500, LDAP and XML Enabled Directory (XED) standards, ViewDS provides a rich set of features offering many benefits for directory users and administrators. The major features of ViewDS are outlined in the following topics:

Improved user support

ViewDS makes users' access to a directory easier by providing a web-based client and by supporting advanced approximate matching.

Web-based client

The ViewDS web-based client, Access Presence, reduces the time required for implementation and support by providing client functionality that can be customized
to your requirements.

This functionality includes:

- Organization charts
- Self-service portals
- Reporting interfaces
- Certificate management
- Chinese language approximate matching

Approximate matching

Users are often imprecise when they search a directory – they might, for example, misspell or mistype names, or use acronyms or abbreviations. ViewDS delivers superior approximate matching that supports a range of strategies and provides better service to users:

- phonetic matching – for example, the search criteria ‘pane’ would match ‘payne’
- typing correction – ‘Dircetor’ would match 'Director'
- stem matching – ‘optics’ would match ‘optical’
- synonym matching – 'Bob' would match 'Robert', and 'road' would match 'street'
- abbreviation matching – 'NSW' would match 'New South Wales'

Additionally, specialized indexes are used to rapidly evaluate approximate matches.

Data represents your real-world hierarchy

ViewDS minimizes the time, resources and planning required to make large-scale changes to a Directory Information Tree (DIT).

Directory Information Tree

Directory entries are arranged in a hierarchy called the DIT. A directory is of most use when its DIT mirrors a hierarchy in the real-world, such as the structure of a company, health trust or government department.

Real world hierarchies, however, tend to change very frequently – for example, departments can be formed, dissolved, moved, merged or split. This can lead directory administrators to flattening and simplifying the DIT to avoid the complexity of maintaining it. Unfortunately, the result can be little more than a list of entries, rather than a resource that helps people understand the structure of their organization.

ViewDS overcomes this by providing seamless support for changes to complex hierarchies. Directory administrators can change a DIT simply and rapidly so that it always accurately reflects the real-world structure of their organization.

Move and rename

Unlike many directories, ViewDS allows DIT entries with subordinates (non-leaf entries) to be moved or renamed, while looking after all the associated links such as ‘managed by’ and ‘manager of’ relationships. ViewDS can be configured to enforce various referential integrity constraints.

XACML integration

ViewDS leverages its XML capabilities to allow XACML policy to be defined that conforms to XACML Version 3.0. This standard describes a language for expressing access controls and a protocol for requesting access-control decisions, both written in XML.

ViewDS includes the XACML Access Control scheme, which allows XACML policy to be applied to the ViewDS directory.

ViewDS Access Sentinel extends the capability to allow XACML policy to be applied to applications external to ViewDS permitting unified access control across disparate applications and data sources, improving the user experience and reducing administration and support. For more information about Access Sentinel see the ViewDS Access Sentinel Installation and Reference Guide.

Flexible data search

The ViewDS component-matching rules allow users to search specific parts of an attribute with a complex syntax – for example, digital certificates, XML documents and certificate-revocation lists.

To illustrate, consider a Human Resources department that stores employees' resumes as XML documents. Component matching would allow a specific area of
the resumes to be searched individually, and therefore, efficiently. For example, a user might search on just the 'qualifications' component of the resumes to find employees with a qualification in nuclear physics.

Without component matching, an application developer would generally need to either scan an entire directory or filter search results in order to find the required data. Both options are inefficient and slow.

PKI matching rules

X.509 certificates can be stored by most directories, but few support matching rules for the X.509 Public Key Infrastructure (PKI) attribute syntaxes. Consequently, environments with large certificate-revocation lists can suffer unacceptable performance.

ViewDS supports the PKI matching rules, and also supports component matching for the PKI syntaxes. A PKI application that uses ViewDS will process certificates faster and with less effort. ViewDS also supports strong authentication with X.509 certificates for client-to-server and server-to-server authentication.

Windows-based management tool

The ViewDS Management Agent is a Windows-based application that makes managing Directory System Agents (DSAs) simpler and more efficient.

Certificate lookup service

ViewDS Access Proxy provides a certificate lookup service that complies with TSCP specifications. For more information about Access Proxy, see the ViewDS Access Proxy Installation Guide.

Reliability with scalability

ViewDS is designed to accommodate 'mission-critical' directory applications and is designed for continuous operation. During routine maintenance – such as backing up the database and check-pointing update logs – ViewDS continues to process queries.

ViewDS is designed to be scalable:

- A single ViewDS server can accommodate many millions of entries.
- There are no restrictions on the number of entries, size or number of attributes, depth of the DIT or number of connected users (other than those imposed by the host's operating system and hardware).
- ViewDS has an optimized tool for fast loading of entries. During bulk loading, the difference between the load-speed for the first and last entries loaded is negligible.
- ViewDS restarts rapidly after a power failure, irrespective of the number of entries.

Integrates with your existing technology

The ViewDS standards-based architecture ensures seamless communication with third-party directory clients and servers:

- Implements OSI Stack (including RFC1006 over TCP/IP) and X.500 Internet Directly Mapped Protocol (IDMP).
- Supports LDAP, XLDAP, IDMP, XIDMP, SPML, SNMP and HTTP access.

Additionally, ViewDS components communicate using an efficient proprietary protocol running over TCP/IP.

Installing ViewDS

This section describes how to install ViewDS.

The installation includes the ViewDS server (Directory System Agent and Remote Administration Service) and the ViewDS Management Agent, plus other components that require additional licensing (including Access Sentinel).

ViewDS installation scenarios

ViewDS can be installed in a variety of ways to suit your deployment needs. To illustrate this, a number of common deployment scenarios are described below. Follow one of these installation paths, adapt one to suit your needs, or contact your reseller for specific advice about your installation.

Complete install on single Windows server

In this scenario all the ViewDS components are installed on the same Windows server. This deployment scenario has the quickest and simplest installation path since all the required steps (including the installation and configuration of prerequisite software for the ViewDS Management Agent (VMA) and Access Presence, the installation of temporary keys, license key installation and the creation of a connection between the ViewDS server and the VMA) are handled automatically by the installer:

1. Install the ViewDS suite on Windows

> The ViewDS suite installer will install .NET Framework 3.5 and Open XML SDK 2.0 (required by the VMA) and enable the installed version of IIS (required by Access Presence) if supported versions of these applications are not already installed or enabled on the chosen Windows machine.

ViewDS server and Access Presence on Linux with remote VMA

In this scenario the ViewDS server and Access Presence are installed on a Linux server and the VMA is installed on Windows:

1. Install ViewDS server on Linux
2. Install ViewDS Management Agent on Windows
3. Install credentials

4. Create connection between ViewDS server and ViewDS Management Agent
5. Enter license information

> Additionaly, if you wish to use Access Presence you must install and configure an Apache HTTP server on the chosen Linux machine.

ViewDS server and Access Presence on Windows with remote VMA

In this scenario the ViewDS server and Access Presence are installed on one Windows server and the VMA is installed on another. This type of deployment would be suitable if you want to install the VMA on an administrator's desktop PC, allowing them to remotely manage multiple ViewDS server instances across a number of platforms:

1. Install ViewDS server on Windows
2. Install ViewDS Management Agent on Windows
3. Install credentials
4. Create connection between ViewDS server and ViewDS Management Agent
5. Enter license information

> Additionaly, if you wish to use Access Presence you must install and configure a supported HTTP server on the chosen Windows machine.

ViewDS server on Linux with remote Access Presence and VMA

In this scenario the ViewDS server is installed on a Linux server, Access Presence is installed on one Windows server and the VMA is installed on another. This type of deployment might be suitable for an internet facing Access Presence instance, where the Access Presence machine is in the DMZ but the ViewDS server machine is within the internal network:

1. Install ViewDS server on Linux
2. Install ViewDS server (including Access Presence) on Windows
3. Configure Access Presence on the remote host
4. Install ViewDS Management Agent on Windows
5. Install credentials
6. Create connection between ViewDS server and ViewDS Management Agent
7. Enter license information

Upgrading from a previous version

If you are an existing customer, perform one of the following tasks to upgrade from a previous version of ViewDS or View500:

- Upgrading from a previous version of ViewDS
- Upgrading from View500

Upgrading from a previous version of ViewDS

To upgrade from ViewDS version 7.0, 7.1, 7.2, 7.3:

- Dump database and back up
- Uninstall the existing version and install ViewDS
- Load the database

> Before starting the upgrade, check the ViewDS release notes for any known issues that may hinder the process.

Dump database and back up

> As a result of changes made to the way in which passwords are encrypted in dumps and logs in ViewDS 7.4, it is essential that you know the key used to encrypt passwords in the files that you are about to dump. This key is available through the ViewDS Management Agent, on the **Configuration** tab in the **Runtime Settings** pane. If no key is included on the **Runtime Settings** pane, then the default ViewDS key - MePh2P - is used. You will need this key when loading the dumped files into ViewDS 7.4.

> As a result of changes to the format of the ViewDS license key made in ViewDS 7.4, you will not be able to use your existing license key when you upgrade from a previous version. Before upgrading you must contact your ViewDS vendor with your old license in order to obtain a replacement key. See the License requirements topic for further details.

To dump the database of your existing installation and back it up:

1. From the ViewDS Management Agent, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Graphical View** button.
4. Right-click the **Database** (DB) icon and then click **Dump**. The dump files are written to the dump directory.
5. Finally, set the logging level for the error log to **debug** to facilitate troubleshooting later:
    a. At the bottom of the left pane, click **Server View**.
    b. In left pane, click the server.
    c. In the right pane, click the **Configuration** tab and then click **Operational**. This screen includes Error-level parameter.
    d. In the value column for the **Error level**, select **debug**.
    e. At the bottom of the screen, click the **Set** button.
6. Check that the dump files have been created in the dump directory and back them up. The location of the dump directory is set by the configuration-file parameter `dumpdir`, which by default is `${VFHOME}/dump` (Linux or Solaris) or `%VFHOME%\dump` (Windows).

    where ${VFHOME} or %VFHOME% is the install folder of old version of ViewDS.
7. Back up your installation directory by archiving it and its sub-directories (this includes the licence information stored in `${VFHOME}/setup/config`).
8. Perform the steps below to uninstall the existing version and install ViewDS.

Uninstall the existing version and install ViewDS

To uninstall the existing version and install ViewDS:

1. From the ViewDS Management Agent's host PC, open the **Windows Control Panel** and uninstall the **ViewDS Management Agent**.
2. Perform the task Uninstalling the DSA and RAS.
3. Perform the tasks outlined for your chosen installation path in the Installation scenarios topic.
4. Confirm the Configuring the DSA, particularly the runtime parameters.
5. Perform the steps below to load the database.

Load the database

To load your database into the new installation:

1. Copy all the previously backed up files that are installation-specific to the appropriate locations in the new ViewDS installation.

   These files include, for example:

   - configuration file (by default, `${VFHOME}/setup/config`)
   - Access Presence files (by default, located in `${VFHOME}/webdir/`)
   - administration support scripts
   - certificates

2. Remove the following deprecated parameters (if they exist) from the configuration file (by default, `${VFHOME}/setup/config`):

   - `xentries`
   - `xexpiry`
   - `xhostid`
   - `xkeybase`
   - `xlicensekey`
   - `xlimit`
   - `xschema`
   - `xsearches`
   - `xdsa`
   - `xpdua`
   - `xsduasync`
   - `xwebdua`
   - `xenforce`
   - `xtscpproxy`
   - `xpdp`

3. Copy the dump files of your existing database into the new ViewDS dump directory. The location of the dump directory is set by the configuration-file parameter `dumpdir`, which by default is:
   - `${VFHOME}/dump` (Linux or Solaris)
   - `%VFHOME%\dump` (Windows)

4. Enter the following on the command line to stop the RAS and the DSA:

   ```
   ras stop
   ```

5. Then enter this command to load the dumped database files:

```
vfload -K "encryption key" -m dump/dib.*
```

Where the `encryption key` is the key used to encrypt passwords in dump files in your old version of ViewDS. See the note at the beginning of Dump database and backup for details.

You may need to address any schema inconsistencies before the dump files can be loaded successfully. Details of any schema parsing problems will be reported in the error log (if the logging level is set to **debug**).

6. Finally to restart the RAS and the DSA use the following command:

   ```
   ras
   ```

7. Start the **ViewDS Management Agent**.
8. Click **Server View**
9. In the left pane, click the server.
10. In the right pane, click the **Status** tab followed by the **Error Log** tab.
11. Review the contents of the error log.

Upgrading from View500

To upgrade from View500:

- Dump database and back up
- Uninstall View500 and install ViewDS
- Load the database

> Before starting the upgrade, check the ViewDS release notes for any known issues that may hinder the process.

Dump database and back up

> As a result of changes made to the way in which passwords are encrypted in dumps and logs in ViewDS 7.4, it is essential that you know the key used to encrypt passwords in the files that you are about to dump. This key is available in the file `VFHOME/general/deity` when the DSA is running. The key is the string that appears after the comma in this file. So, for example, the file might contain the following: 745403800,MePh2P and MePh2P is the key.

As a result of changes to the format of the ViewDS license key made in ViewDS 7.4, you will not be able to use your existing license key when you upgrade from a previous version. Before upgrading you must contact your ViewDS vendor with your old license in order to obtain a replacement key. See the Licening requirements topic for futher details.

To dump the database of your existing installation and back it up:

1. Enter the following at the command line to stop the DSA and prevent any further database transactions:

   ```
   dsa stop
   ```

2. Dump the database:

   ```
   vfload -c dump
   ```

3. Check that the dump files have been created in the dump directory and back them up. The location of the dump directory is set by the configuration-file parameter dumpdir, which by default is:

   - `${VFHOME}/dump` (Linux or Solaris)
   - `%VFHOME%\dump` (Windows)

   where `${VFHOME}` or `%VFHOME%` is the location of the previous version.

4. Back up your installation directory by archiving it and its sub-directories (this includes your licence information stored in `${VFHOME}/setup/config`).
5. Perform the steps below to uninstall the existing version and install ViewDS.

Uninstall View500 and install ViewDS

To uninstall View500 and install ViewDS:

1. From the ViewDS server's host, do one of the following:

   - For Windows, uninstall the DSA service (`dsa -u`) and uninstall View500 from the Windows **Control Panel**.
   - For Solaris or Linux, stop the View500 processes.

2. If implemented, remove all references to the Web Admin interface from your web server. (This application is not supported in ViewDS.)
3. Perform the tasks outlined for your chosen installation path in the Installation scenarios topic.

4. Confirm the Configuring the DSA, particularly the runtime parameters.
5. Perform the steps below to load the database.

Load the database

To load your database into the new installation:

1. Copy all the previously backed up files that are installation-specific to the appropriate locations in the new ViewDS installation.

   These files include, for example:

   - configuration file (by default, `${VFHOME}/setup/config`)
   - Access Presence files (by default, located in `${VFHOME}/webdir/`)
   - administration support scripts
   - certificates

2. Remove the following deprecated parameters (if they exist) from the configuration file (by default, `${VFHOME}/setup/config`):

   - `Webaddauxoctemplate`
   - `xentries`
   - `xexpiry`
   - `xhostid`
   - `xkeybase`
   - `xlicensekey`
   - `xlimit`
   - `xschema`
   - `xsearches`
   - `xdsa`
   - `xpdua`
   - `xsduasync`
   - `xwebdua`
   - `xenforce`
   - `xtscpproxy`
   - `xpdp`

3. Copy the dump files of your existing database into the ViewDS dump directory. The location of the dump directory is set by the configuration-file parameter `dumpdir`, which by default is:

- ${VFHOME}/dump (Linux or Solaris)
- %VFHOME%\dump (Windows)

4. Start the **ViewDS Management Agent** and set the logging level for the error log to **debug** to facilitate troubleshooting later:
   a. At the bottom of the left pane, click **Server View**.
   b. In left pane, click the server.
   c. In the right pane, click the **Configuration** tab and then click **Operational**. This screen includes Error-level parameter.
   d. In the value column for the **Error level**, select **debug**.
   e. At the bottom of the screen, click the **Set** button.

5. Enter the following on the command line to stop the RAS and the DSA:

   ```
   ras stop
   ```

6. Then enter this command to load the dumped database files:

   ```
   vfload -K "encryption key" -m dump/dib.*
   ```

   Where the `encryption key` is the key used to encrypt passwords in dump files in your old version of ViewDS. See the note at the beginning of Dump database and backup for details.

   You may need to address any schema inconsistencies before the dump files can be loaded successfully. Details of any schema parsing problems will be reported in the error log (if the logging level is set to **debug**).

7. Finally to restart the RAS and the DSA use the following command:

   ```
   ras
   ```

8. Restart the **ViewDS Management Agent**.
9. Click **Server View**
10. In the left pane, click the server.
11. In the right pane, click the **Status** tab followed by the **Error Log** tab.
12. Review the contents of the error log.

Installing the ViewDS suite on Windows

An installer is provided which allows you to install all the ViewDS components (the ViewDS server, Access Presence and the ViewDS Management Agent) on the same Windows machine. The installer handles all

the required steps automatically including the installation and configuration of prerequisite software for the ViewDS Management Agent (VMA) and Access Presence, the installation of temporary keys, license key installation and the creation of a connection between the ViewDS server and the VMA, making this the quickest and easiest way to get ViewDS up and running.

To install the ViewDS suite on Windows:

1. Log in as a user with Administrator privileges.
2. Run the ViewDS suite installer supplied on the ViewDS installation media. The ViewDS Suite Setup Wizard is displayed.
3. Click **Next**.
4. Scroll through to read and check the box to accept the terms of the license agreement, then click **Next**.
5. Accept the default ViewDS install folder or provide an alternative install location, then click **Next**.
6. Optionally, uncheck the boxes of any components you do not wish to install, then click **Next**. The following components are listed and all of them are installed by default:
    - ViewDS Server
    - ViewDS Access Presence
    - ViewDS Management Agent
7. Paste the text of your license key into the box, if you have one, then click **Next**.

This screen is presented only if you are installing the ViewDS Server.

If you install ViewDS without a license key, then you will be able to launch the VMA but you will not be able to create a connection to the ViewDS server. To complete an installation in these circumstances, you should provide license key information as described in the topic Entering license information.

8. Choose which ViewDS shortcuts you want the installer to create by checking the appropriate boxes, then click **Next**. By default shortcuts are created for the VMA on the desktop and in the Start menu All Programs folder under ViewDS Suite. In addition,shortcuts are also created in this folder to the default Access Presence page containing the ViewDS example dataset, Deltawing and the ViewDS website, ViewDS Identity Solutions.
9. Optionally, uncheck the **Install the default certificate** box if you do not want the installer to install default PKI certificates.

This option is presented only if you are installing the VMA.

Default certificates are required in order to create a connection between the ViewDS server and the VMA.

10. Click **Install**.

The ViewDS suite installer will install .NET Framework 3.5 and Open XML SDK 2.0 (required by the VMA) and enable the installed version of IIS (required by Access Presence) if supported versions of these applications are not already installed or enabled on the chosen Windows machine.

11. If default certificates are being installed, then the Certificate Import Wizard will be displayed during the installation:
    a. Click **OK**.
    b. Click **Next**.
    c. The name of the default certificate is displayed on the File to Import screen. Click **Next**.
    d. No password is required for the private key on the Password screen. Click **Next**.
    e. On the Certificate Store screen, the automatically selected certificate store should be specified. Click **Next**.
    f. Click **Finish**, then click **OK**.
12. Optionally, uncheck **Configure Management Agent** to prevent the installer from configuring a connection between the VMA and the ViewDS server.

This option is presented only if you installed the ViewDS Server.

13. Optionally, uncheck **Run Management Agent** to prevent the installer from launching the VMA.

This option is presented only if you installed the VMA.

14. Otherwise, click **Close** to configure the connection between the VMA and the ViewDS server and then launch the VMA.

Installing the ViewDS server

This subsection describes requirements, pre-installation tasks, and how to install the ViewDS server (DSA and RAS). The following are also installed according to your licence: Access Presence, Access Sentinel, Stream DUA and other tools.

The installation includes a demonstration directory, Deltawing.

License requirements

The ViewDS license key has several components that enable each of the following:

- ViewDS server DSA
- Access Sentinel
- Access Proxy

Obtaining a license

To obtain a license key, supply your ViewDS vendor with host-specific information for the computer on which ViewDS will be installed. Your vendor will return a license key that you will need during installation.

Platform	Command	Returns host-specific information
Linux	ifconfig –a	Hardware address of any of the Ethernet interfaces reported by this command. This address is usually represented as a colon separated list of six hexadecimal fields.
Solaris	sysdef –h	Host identifier of this host.
Windows	ipconfig /all	Physical address of the Ethernet adaptor reported by this command. This address is usually six hexadecimal fields, each separated by a dash. For example: 00-22-BA-7D-92-DE

If you are upgrading

As a result of changes to the format of the ViewDS license key made in ViewDS 7.4, you will not be able to use your existing license key when you upgrade from a previous version. Instead you must contact your ViewDS vendor with your old license in order to obtain a replacement key. Your old license key is stored in the configuration file located by default in either:

- `${VFHOME}/setup/config` (Linux and Solaris)
- `%VFHOME%\setup\config` (Windows)

Where `${VFHOME}` or `%VFHOME%` is the ViewDS install directory.

ViewDS server requirements

This subsection describes the following requirements for the ViewDS server:

- Platform requirements
- Memory requirements
- Disk space requirements

Platform requirements

To install ViewDS DSA, you need one of the following:

- 64-bit Sun SPARC server running Solaris 10 or later
- 32 or 64-bit Intel running Windows Server 2003 or later
- 32 or 64-bit Intel running RedHat Enterprise Linux 5 or later

Other vendors' platforms may also be suitable – contact your ViewDS vendor for further information.

Memory requirements

For optimum performance there should be sufficient real memory to run the DSA process suite, and to provide the DSA with a cache sufficient to hold all entries with subordinates (non-leaf entries) in memory.

Recommendation for a medium size directory

The size of processes varies with the processor type, and the software and directory configuration. A typical minimum for a medium size organization is 32 MB, with 64 MB preferred. Additional memory should also be allocated to the DSA disk cache.

If this cache is as large as the database, the entire database will reside in the cache and disk accesses minimized. This configuration gives optimum performance and is the recommended configuration for smaller databases.

Recommendation for a larger directory

For a very large database (millions of entries), the performance difference between having all non-leaf entries and indexes in cache and all entries and indexes in cache is too small to justify a large cache. The recommendation is for sufficient memory to hold only all non-leaf entries and indexes in cache, perhaps 10% of the database size.

Disk space requirements

The disk space required depends on the number of entries, the amount of data they hold, and the amount of indexing configured. There must be enough space to hold the database and the dumped database.

A typical minimum requirement is approximately 30 MB plus 2 KB for each entry. This space must be available after providing for operating system requirements including swap space. This implies actual disk capacities in the order of 300 MB for medium organizations and 500 MB for larger organizations.

Disk reliability and redundancy measures, such as implementing a RAID-1 or RAID-5 disk array, should also be considered.

Access Presence requirements

Access Presence (webdua.cgi) is a web-based client included in the installation of the ViewDS server. If you intend to implement Access Presence, a web server is required on either the same or a different host to the ViewDS server. The supported web servers are Apache HTTP Server and Microsoft Internet Information Services (IIS) Versions 6, 7 and 8.

Installing the ViewDS server

Solaris or Linux

To install the ViewDS server with the default configuration:

1. Create a Solaris or Linux account for the ViewDS System Administrator.

   The ViewDS processes on Solaris and Linux run under a single account name (the ViewDS System Administrator), which should have an account with a login name indicative of the ViewDS application (for example, viewds). By default, the locations of the ViewDS directories are relative to this user's home account.

   The administrator does not need to be a super-user.

2. Log in as the ViewDS System Administrator and move to the directory where ViewDS should be installed. This will be the ViewDS installation directory, referred to as `${VFHOME}`.

   ViewDS will be installed as a collection of subdirectories below `${VFHOME}`.

3. From the installation media, select the appropriate tar file for your platform and un-tar it. The ViewDS directories and files are created in the current directory; one of the files is vfinstall in the install directory.

4. Run the script `vfinstall`. You will be prompted to set the `VFHOME`, `MANPATH`, `PATH` and `LD_LIBRARY_PATH` environment variables.
5. Install the ViewDS Management Agent.

Windows

To install the ViewDS server with the default configuration:

1. Log in as a user with Administrator privileges.
2. Run the ViewDS suite installer supplied on the ViewDS installation media. The ViewDS Suite Setup Wizard is displayed.
3. Click **Next**.
4. Scroll through to read and check the box to accept the terms of the license agreement, then click **Next**.
5. Accept the default ViewDS install folder or provide an alternative install location, then click **Next**.
6. Uncheck the box for **ViewDS Management Agent**, then click **Next**.
7. Paste the text of your license key into the box, if you have one, then click **Next**.

> If you install ViewDS without a license key, then you will not be able to create a connection to the ViewDS server. To complete an installation in these circumstances, you should provide license key information as described in the topic Entering license information.

8. Choose which ViewDS shortcuts you want the installer to create by checking the appropriate boxes, then click **Next**.
9. Click **Install**.
10. Optionally, uncheck **Configure Management Agent** to prevent the installer from configuring a connection between the VMA and the ViewDS server.
11. Otherwise, click **Close** to configure the connection between the VMA and the ViewDS server.

Uninstalling the DSA and RAS

To upgrade from ViewDS 7.0, 7.1, 7.2 or 7.3 to ViewDS 7.4 you must uninstall the old DSA and RAS before running the ViewDS 7.4 installer.

Solaris or Linux

To uninstall on a Solaris or Linux platform:

1. Stop the RAS if it is running:

   ```
   ras stop
   ```

   This also stops the DSA.

2. Delete the ViewDS installation directory and the ViewDS administration account.

Windows

To uninstall on a Windows platform:

1. From the command line, stop the RAS (which also stops the DSA) and remove it as a service:

   ```
   ras stop
   ```

   ```
   ras -u
   ```

2. Uninstall ViewDS from the Windows **Control Panel**. This removes all unmodified files in the `%VFHOME%` folder.

Installing the ViewDS Management Agent

ViewDS Management Agent is a Windows-based application that allows you to manage the status of one or more local or remote DSAs and access their configuration parameters, log files, directory data, schema, knowledge and access controls.

This section describes the requirements for the ViewDS Management Agent and how to install it.

ViewDS Management Agent requirements

The ViewDS Management Agent can be installed on any 32 or 64 bit Intel system running a Windows operating system. In addition, it requires the Windows .NET Framework Version 3.5, Open XML SDK 2.0 and Visual C++ libraries.

Installing the ViewDS Management Agent

The ViewDS Management Agent is a Windows-based application that allows you to manage the status of one or more ViewDS servers and access their configuration parameters, log files, directory data, schema, knowledge and access controls.

To install the ViewDS Management Agent:

1. Log in as a user with Administrator privileges.
2. Run the ViewDS suite installer supplied on the ViewDS installation media. The ViewDS Suite Setup Wizard is displayed.
3. Click **Next**.
4. Scroll through to read and check the box to accept the terms of the license agreement, then click **Next**.
5. Accept the default ViewDS install folder or provide an alternative install location, then click **Next**.
6. Uncheck the boxes for **ViewDS Server** and **ViewDS Access Presence**, then click **Next**.
7. Choose which ViewDS shortcuts you want the installer to create by checking the appropriate boxes, then click **Next**.
8. Optionally, uncheck the **Install the default certificate** box if you do not want the installer to install default PKI certificates.

> Default certificates are required in order to create a connection between the ViewDS server and the VMA.

9. Click **Install**.

> The ViewDS suite installer will install .NET Framework 3.5 and Open XML SDK 2.0 if these are not already installed on the chosen Windows machine.

Installing credentials

ViewDS uses SSL/TLS connections and certificate-based authentication. This ensures secure communications between the ViewDS Management Agent and the ViewDS server (DSA and RAS) and between the DSA and RAS.

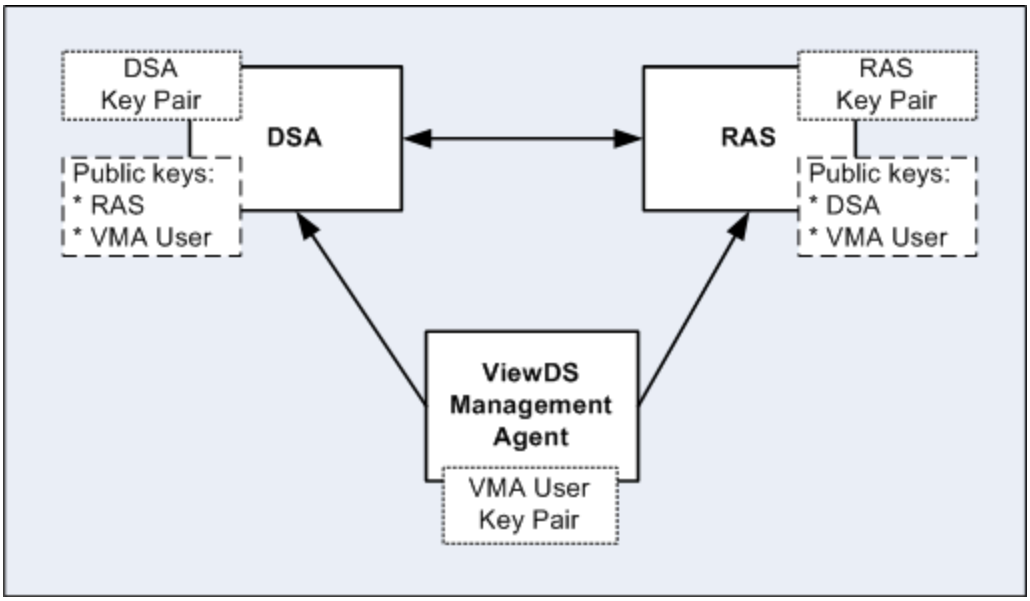


Figure 4: ViewDS certificates

ViewDS is distributed and can be installed with a default key pair – public key and private key. However, it is strongly recommended that you obtain and install your own public–private key pairs for each DSA and RAS. (Every DSA in a distributed or replicated environment must have a certificate with a unique subject name.)

Each user of the ViewDS Management Agent must also have a key pair. Otherwise, the ViewDS Management Agent will not be able to connect to either the DSA or RAS.

Obtaining SSL server certificates

There may already be processes in place within your organization to generate certificates. If not, new SSL server certificates can be purchased online from a Certificate Authority or generated using an open source tool (such as OpenSSL). If required, please contact your ViewDS vendor for further advice.

The private keys for the DSA and RAS must be in either a PKCS#8 (BER) or PKCS#12 (DER) file.

To install key pairs see Installing new key pairs.

Installing new key pairs

Installing new key pairs involves:

- Installing a key pair for a ViewDS Management Agent user
- Installing a new key pair for the DSA and RAS

Installing a key pair for a ViewDS Management Agent user

To install a public and private key pair from a PKCS#12 format file:

1. On the ViewDS Management Agent's host computer, install the user's keys:
    1. Copy the PKCS#12 file to the ViewDS Management Agent's host computer.
    2. Double-click the PKCS#12 file. A certificate window is displayed.
    3. Follow the instructions on the screen to import the certificate.
2. Export the public key certificate:
    1. Start the ViewDS Management Agent.
    2. From the **File** menu, click **New Connection**. The New Connection window is displayed.
    3. In the **Certificate for DSA** box, click the certificate that you have just installed.
    4. Click the **View** button. The Certificate window is displayed.
    5. Click the **Details** tab.
    6. In the **Field** column, click **Public Key** and then click the **Copy to File** button. The Certificate Export Wizard is displayed.
    7. Follow the instructions on the screen and export the public key to a DER file. When the export has completed, the Wizard closes but the New Connection window is still displayed.
    8. In the New Connection window, click the **Cancel** button.
3. On the DSA's host, copy the exported public-key certificate to the following location `${VFHOME}/setup/trusted` or `%VFHOME%\setup\trusted` (`VFHOME` is the ViewDS installation path).

    This is the default location for public keys stored by both the DSA and RAS. Each location can be modified through settings on the File System tab of the ViewDS Management Agent: 'Directory of DSA administrator certificates' and 'Directory of remote administration service administrator certificates'.

4. Follow the steps below to connect to the ViewDS server.

Connecting to the ViewDS Server

1. From the main menu of the ViewDS Management Agent, click **File** followed by **New Connection**. The New Connection window is displayed.
2. In the New Connection window, complete the following details:

- **Host** – enter the host name or IP address of the DSA's host.
- **DSA Port** – enter the port number to connect to on the host for the DSA (by default, 3000). The RAS Port box is automatically populated (by default, 3018).
- **Certificate for DSA** – select the certificate used by the ViewDS Management Agent to connect to the DSA. The Certificate for RAS box is automatically populated with the same certificate.
- **Connection Name** – enter the label displayed by the ViewDS Management Agent for this connection (for example, 'Development DSA').

3. Click the **Connect** button. A message window is displayed to say that the DSA will not start without a licence key.
4. Click **OK**. A server icon is now displayed in the left pane.

Installing a new key pair for the DSA and RAS

To install a new key pair for the DSA and RAS:

1. From the ViewDS Management Agent, click the server icon in the left pane.

   Before installing a new key pair, the DSA should not be running. However, for a new installation, the DSA will not be running because you will not have entered its licensing information at this stage.

2. In the right pane, click the **Configuration** tab followed by the **File System** tab.

   The tab includes the following settings:

   - **DSA public key (certificate)** – identifies the name and location of the public key on the DSA's host. (This setting corresponds to dsacertificate in the DSA's configuration file.)
   - **DSA private key** – identifies the name and location of the private key on the DSA's host. (This setting corresponds to dsaprivkey in the configuration file.)
   - **DSA private password** – identifies the file containing the clear-text password required to decrypt the DSA's private key.
   - **Directory of remote administration service administrator certificates** – identifies the location for public keys used by the RAS. (This setting corresponds to rastrusted in the configuration file.)

By default, the configuration file is `${VFHOME}/setup/config` or `%VFHOME%\-setup\config`.

3. On the DSA's host, copy the new private key to the location set by DSA private key. By default, `${VFHOME}/setup` or `%VFHOME%\setup`.

   The file should be read-only to the owner of the DSA process.

4. Copy the new public key to the locations set by:

   - **DSA public key (certificate)** – by default, `${VFHOME}/setup` or `%VFHOME%\setup`
   - **Directory of remote administration service administrator certificates** – by default, `${VFHOME}/setup/trusted` or `%VFHOME%\setup\trusted`

5. If the private key is encrypted with a password, enter the password into the file identified by **DSA private password**. By default, `${VFHOME}/general/ keyaccess` or `%VFHOME%\-general\keyaccess`.
6. From the ViewDS Management Agent, double-click the Value cell for the **DSA private key** setting. The cursor is displayed in the cell.
7. Modify the value to the file name of the new private key (also modify the path if required).
8. Set the values of **DSA public key (certificate)** and **Directory of remote administration service administrator certificates** to the file name of the new public key.
9. At the bottom of the screen, click the **Set** button.
10. To install a new key pair for the RAS, repeat this task from step 2 using the following settings:

    - **Remote administration service public key (certificate)** – identifies the name and location of the RAS's public key. (This setting corresponds to rascertificate in the DSA's configuration file.)
    - **Remote administration service private key** – identifies the name and location of the RAS's private key. (This setting corresponds to rasprivkey in the configuration file.)
    - **Remote administration service private password** – identifies the file containing the clear-text password required to decrypt the RAS's private key.
    - **Directory of DSA administrator certificates** – identifies the location of public keys used by the DSA. (This setting corresponds to dsatrusted in the configuration file.)

11. Perform the task Entering licence information.

Installing a temporary key pair

You can implement a working, albeit temporary and non-secure, system by:

- using the default key pairs supplied with the DSA and RAS; and
- using the same default key pair used by the RAS for a user of the ViewDS Management Agent. A PKCS#12 file of the default RAS public and private key pair is provided for this purpose.

> This is a temporary measure. Using the key pairs distributed with ViewDS in a production environment is not recommended. This is because they are available to all customers of ViewDS and cannot be considered secure.

To install the RAS's key pair for a user of the ViewDS Management Agent (this task may vary slightly according to the operating system):

1. In the folder where ViewDS Management Agent is installed (by default, C:\Program Files\ViewDS Management Agent), double-click the file ras.p12. A certificate window is displayed.
2. Follow the instructions on the screen to import the certificate.

   > A password for the private key is not required.

3. When installation is complete, follow the steps below to connect to the ViewDS server.

Connecting to the ViewDS server

To connect the ViewDS Management Agent to the ViewDS server:

1. Start the ViewDS Management Agent. The application starts with the New Connection window open. (If this window is not displayed, click **File** from the main menu followed by **New Connection**.)
2. In the New Connection window, complete the following details:
   - **Host** – enter the host name or IP address of the DSA's host.
   - **DSA Port** – enter the port number to connect to on the host for the DSA (by default, 3000). The RAS Port box is automatically populated (by default, 3018)
   - **Certificate for DSA** – select the RAS certificate. The Certificate for RAS is automatically populated with the same certificate.
   - **Connection Name** – enter the label displayed by the ViewDS Management Agent for this connection (for example, 'Development DSA').
3. Click the **Connect** button. A message window is displayed to say that the DSA will not start

without a licence key.

4. Click **OK**. A server icon is now displayed in the left pane.

Entering licence information

You will need licence information (see License requirements) to perform this task.

To enter licence information using the VMA and start the DSA:

1. In the left pane of the ViewDS Management Agent, click the server icon.
2. In the right pane, click the **Configuration** tab.
3. Within the Configuration tab, click the **Licence** tab. An empty Licence screen is displayed.
4. Do one of the following:
    - Open the file containing the licence information, and copy and paste it into the Licence screen.
    - From the Licence screen, click the **Import** button and then select the file containing the licence information.
5. At the bottom of the right pane, click **Set**. The licence information is saved and you are given the option to restart the server.
6. Click the **Yes** button. The DSA restarts.
7. In the right pane, click the **Status** tab followed by the **General** tab. Confirm that the DSA is running and the database is open.

Configuring ViewDS

This section describes the following areas of configuration:

- Configuring the DSA
- Configuring for Access Presence
- Configuring for the XACML framework

Configuring the DSA

This section describes how to use the ViewDS Management Agent to modify the following groups of Directory System Agent (DSA) parameters:

- Runtime settings
- Addresses
- Operational
- Licence

The first group, the runtime settings, can also be modified through the DSA Controller, Stream DUA or ViewDS Fast Load utility (see the ViewDS Technical Reference Guide: Directory System Agent).

The remaining parameters are stored in the DSA's configuration file, which is a text file in the following default location:

- `${VFHOME}/setup/config` (Solaris or Linux)
- `%VFHOME%\setup\config` (Windows)

where `${VFHOME}` or `%VFHOME%` is the location of ViewDS.

Runtime settings

To view or modify the local settings through the ViewDS Management Agent:

1. If your ViewDS installation is licensed for more than just Access Proxy, click Server View at the bottom or the right pane. (When ViewDS is licensed for just Access Proxy, this button is unavailable.)
2. In the left pane, click the server.
3. In the right pane, click the Configuration tab and then click Runtime Settings.

The more important local settings are described below.

DOT threads

DOT (Directory Operation Thread) threads process the requests assigned to them by the DSA.

Each DOT thread that belongs to a DSA handles database queries synchronously. By having multiple DOT threads, the DSA can process queries asynchronously – that is, multiple queries can be processed simultaneously.

There are implications to having different numbers of DOT threads:

DOT threads	Description
3 DOT threads	The default setting. Having more than three DOT threads might improve throughput, but it will be at the expense of memory use.
2 DOT threads	If the system is low on memory, try running with two DOT threads.
1 DOT thread	This setting is appropriate for single-user operation, but it will slow throughput with multiple users. More than one DOT thread is required for replication and is strongly recommended for distributed operations in order to prevent the DSA deadlocking.

Three or four threads will deliver the best performance for a non-distributed DSA with a high query load. To reduce the risk of memory exhaustion, not more than 3 DOTs should be used on a 32-bit deployment. The maximum setting is 5.

Max DOT size

This is the maximum size that a DOT thread can reach before it is restarted.

The size of a DOT thread increases according to the size of each query it receives. If a DOT receives a query that takes it above the 'Max DOT size', it will handle the query and then restart. It is more efficient to have a 'Max DOT size' that avoids this scenario.

The default is 20 MB.

Max. updates

This is the number of DOT threads that process update operations simultaneously.

The maximum number of updates:

- cannot exceed the number of DOT threads.
- should be less than the number of DOT threads to ensure that some DOT threads are always available for queries. Otherwise, during heavy updating,
  the directory may be too slow in responding to queries.
- should be set to 1 for normal operation.
- should be set to 0 to disable updates to the database.
- should be greater than 1 for a DSA in a replication agreement. (For distributed operations, this setting avoids deadlocks while processing a chained update operation. For replication, this setting avoids DAP/LDAP update operations
  having to wait for considerably longer than normal while replication updates
  are processed.)

Safe size

The safe file is a database recovery log. It contains details of database transactions that are waiting to be committed. After a crash or improper shutdown, the DSA uses the safe file to recover the database before reopening it.

The safe file should be set to a size that will accommodate the largest anticipated transaction. Even though the file grows to accommodate larger transactions, it is advisable to pre-allocate disk space to ensure efficient performance.

A larger size results in faster database writes, but slower database restarts; and a smaller size results in slower writes, but faster restarts. The minimum size is 1 MB, and the default is 8 MB. (The size for the demonstration directory, Deltawing, is 2 MB.)

Cache size

Sets the size of the memory cache used by the database. The larger the cache, the less the database will need to access the disk, and the faster the response time. The cache should therefore be made as large as possible, ideally to the point where the entire database can be held in memory.

If the cache is too small, there is a serious effect on performance.

Setting the Time limit, Size limit and DAP time-out

These settings apply to clients connected to the DSA. They apply to the ViewDS Management Agent, Access Presence, and any LDAP, XLDAP, DAP or administration DUA.

- Time Limit – the DSA's time limit (in seconds) for a DUA user's read, compare, search and list operations. A normal value is 5 seconds. A value of -1 means there is no time limit.
- Size Limit – the maximum number of entries the DSA will return in response to a search or list operation. The default value is 2000.
- DAP Time-out – the time-out period for an inactive connection to a DUA. When there have been no requests from a DUA for the number of seconds set by this parameter, the connection is unbound. If set to zero, there is no time-out period.

A DUA user can override the Time Limit or Size Limit with a lower value. If, however, the user sets a higher value it is ignored by the DSA.

Addresses

The more important address parameters are described below.

DSA Access Point

The address presented by the DSA to allow a DUA or another DSA to access it.

Modify this parameter if the address required by a DUA or another DSA is different to the address used locally. For example, this might be the case if the DSA is behind a firewall that has Network Address Translation (NAT). The IP address used to access the DSA from an external network would therefore be different to the IP address used locally.

To view or modify the DSA Access Point:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab and then click **General**.

DSA LDAP Address

The DSA listens on this address for native LDAP connections. If the DSA SLDAP Address is also defined, the two should specify different port numbers.

Recommended values are:

- `ldap://localhost:389` (Unix root or Windows)
- `ldap://localhost:{baseport}+6` (all others)

There is no default value.

To view or modify the address settings through the ViewDS Management Agent:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Configuration** tab and then click **Addresses**.

Operational

To view or modify the operational parameters through the ViewDS Management Agent:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Configuration** tab and then click **Operational**.

Error level

The error level determines the minimum level of the errors written to the error log:

- `errors` (includes console error messages)
- `warnings`
- `status` (the default setting)
- `debug`

These are minimum levels. Setting the level to debug will report debug, status, warnings and error messages.

The default is `status`.

Licence

To view the licence parameters through the ViewDS Management Agent:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Configuration** tab and then click **Licence**. The licence information is displayed.

> If you modify the licence information provided by your ViewDS vendor, your installation will not operate correctly.

Configuring for Access Presence

Access Presence (webdua.cgi) is a web-based client that is included in the installation of the ViewDS server.

This section describes how to configure a web server so that Access Presence can access the demonstration directory installed with ViewDS, Deltawing. It also describes how to set up Access Presence on a different host to the one running the DSA. This is sometimes desirable for performance or security reasons.

The supported web servers are Apache HTTP Server and Microsoft Internet Information Services (IIS) Versions 6.0, 7.0 or 8.0.

Overview

To run Access Presence on a different host to the one running the DSA, perform the following task:

- Configuring on a remote host

To run Access Presence when it is installed on the same host as your DSA, perform one of the following tasks:

- Configuring Microsoft Internet Information Services
- Configuring Apache HTTP Server

Configuring on a remote host

To install Access Presence on a different host to the one running the DSA:

1. Install the ViewDS server on the host on which Access Presence will be running (see Installing the ViewDS server).

   The installation of the ViewDS server includes Access Presence, by default.

   You do not need a licence key for the Access Presence host.

2. Optionally, delete the **data**, **load** and **dump** folders, as these folders are not required by Access Presence.

For Solaris or Linux

3. Configure a web server for Access Presence (see Configuring Microsoft Internet Information Services or Configuring Apache HTTP Server).
4. In the configuration file on Access Presence's host, set the dsaaddress to the address of the DSA. By default, the configuration file is: `${VFHOME}/setup/config`.

For Windows

3. Remove the **ViewDS Administration** Windows service, as this is not required for a remote instance of Access Presence.

   > The ViewDS suite installer for Windows configures the IIS web server for Access Presence automatically, so it is not necessary to install or configure a supported HTTP server.

4. In the configuration file on Access Presence's host, set the dsaaddress to the address of the DSA. By default, the configuration file is: `%VFHOME%\setup\config`.

Configuring Microsoft Internet Information Services

> In order for Access Presence to work correctly with IIS 7.0 and above, it is essential that the **CGI** role service is installed when IIS is enabled. Refer to the Microsoft documentation for information about how to do this for your version of IIS. If you use the ViewDS suite installer to enable IIS, then the **CGI** role service is installed automatically.

Overview

To allow Access Presence to connect to the demonstration directory, Deltawing, the following configuration is required for Internet Information Services (IIS):

- Set up the following aliases to Access Presence directories:
  - ViewDS - `%VFHOME%\webdir\`
  - Deltawing - `%VFHOME%\webdir\cgi-bin\`
  - icons - `%VFHOME%\webdir\icons\newicons\`
- Grant the following access rights for the Internet Guest account:
  - all access rights to `%VFHOME%\general`
  - all access rights to `%VFHOME%\print`

    - all access rights to `%VFHOME%\webdir\conf`
    - read-only access to `%VFHOME%\webdir`
- Enable permissions for the `webdua.cgi` process to run

> `%VFHOME%` is the ViewDS install folder – the default is `C:\Documents and Settings\All Users\Application Data\ViewDS Suite` for Windows 2003 or `C:\ProgramData\ViewDS Suite\` for Windows 2008 onwards.

IIS 6, 7 and 8 are support by ViewDS and configuration steps are provided below for each of these versions:

- Configuration for IIS 6.0
- Configuration for IIS 7.0 and IIS 8.0

Configuration for IIS 6.0

To configure an installation of IIS 6.0 dedicated to providing access to Deltawing:

1. Start Internet Information Services Manager.
2. Under the existing **Web Sites** configuration, create a virtual directory called ViewDS as follows:
    a. Right-click the **Default Web Site** folder (or a designated web site folder) and select **New** followed by **Virtual Directory...**. The **Virtual Directory Creation Wizard** is displayed.
    b. Click **Next**. The **Virtual Directory Alias** page is displayed.
    c. In the **Alias** box, enter *ViewDS* and then click **Next**. The **Web Site Content Directory** page is displayed.
    d. Click the **Browse...** button and select the path to `VFHOME\webdir` and then click **Next**. The **Virtual Directory Access Permissions** page is displayed.
    e. Ensure the following permissions are selected: **Read**, **Run scripts**.
    f. Click **Finish**. An icon for ViewDS is displayed below the Default Web Site folder.
3. Create an alias definition for Deltawing:
    a. Right-click the **ViewDS** folder and select **New** followed by **Virtual Directory...**. The **Virtual Directory Creation Wizard** is displayed.
    b. Click **Next**. The **Virtual Directory Alias** page is displayed.
    c. In the **Alias** box, enter *Deltawing* and then click **Next**. The **Web Site Content Directory** page is displayed.
    d. Click the **Browse...** button and select the path to `VFHOME\webdir\cgi-bin` and then click **Next**. The **Virtual Directory Access Permissions** page is displayed.

e. Ensure the following permissions are selected: **Read**, **Run scripts**, **Execute**.

f. Click **Finish**. An icon for Deltawing is displayed below ViewDS.

4. Create an alias definition for icons:

   a. Right-click the **ViewDS** folder and select **New** followed by **Virtual Directory...**. The **Virtual Directory Creation Wizard** is displayed.

   b. Click **Next**. The **Virtual Directory Alias** page is displayed.

   c. In the **Alias** box, enter *icons* and then click **Next**. The **Web Site Content Directory** page is displayed.

   d. Click the **Browse...** button and select the path to `VFHOME\webdir\icons\newicons\` and then click **Next**. The **Virtual Directory Access Permissions** page is displayed.

   e. Ensure the following permissions are selected: **Read**.

   f. Click **Finish**. An icon for icons is displayed below ViewDS.

5. Enable permissions for the webdua.cgi process:

   a. Click **Web Service Extensions**.

   b. Select **All Unknown CGI Extensions** and change the status to **Allowed**.

6. Grant the following access rights for the Internet Guest Account `IUSR_<machine_name>`:

   - Full control on `VFHOME\general`
   - Full control on `VFHOME\print`
   - Full control on `VFHOME\webdir\conf`
   - Read-only on `VFHOME\webdir`

   The above are the default locations, which can be modified through the ViewDS Management Agent.

7. Test the configuration by typing the Deltawing URL into your browser:

   `http://host:port/ViewDS/Deltawing/webdua.cgi`

   where the port is the port at which the web server is listening (typically 80).

Configuration for IIS 7.0 and IIS 8.0

To configure an installation of IIS 7.0 or IIS 8.0 dedicated to providing access to Deltawing:

1. Start Internet Information Services Manager.
2. Under the existing **Sites** configuration, create a virtual directory called *ViewDS*:

   a. Right-click the **Default Web Site** folder and select **Add Virtual Directory...**. The **Virtual Directory** dialog is displayed.

b. In the **Alias** box, enter *ViewDS*.
c. In the **Physical path** box, enter `VFHOME\webdir`.
d. Click **OK**. An icon for ViewDS is displayed below the Default Web Site folder.

3. Create an alias definition for Deltawing:
   a. Right-click the **ViewDS** folder and select **Add Virtual Directory...**. The **Virtual Directory** dialog is displayed.
   b. In the **Alias** box, enter *Deltawing*.
   c. In the **Physical path** box, enter `VFHOME\webdir\cgi-bin`.
   d. Click **OK**. An icon for Deltawing is displayed below the ViewDS folder.
4. Create an alias definition for icons:
   a. Right-click the **ViewDS** folder and select **Add Virtual Directory...**. The **Virtual Directory** dialog is displayed.
   b. In the **Alias** box, enter *icons*.
   c. In the **Physical path** box, enter `VFHOME\webdir\icons\newicons`.
   d. Click **OK**. An icon for icons is displayed below the ViewDS folder.
5. Enable permissions for the webdua.cgi process:
   a. Click the **Default Web Site** folder. The IIS configuration icons are displayed in the middle pane.
   b. In the middle pane, double-click the **Handler Mappings** icon. A list of Handler Mappings is displayed.
   c. In the right pane, click **Add Module Mapping...**. The **Add Module Mapping** window is displayed.
   d. In the **Add Module Mapping** window:
      i. In the **Request path** box, enter **.cgi*
      ii. In the **Module** box, enter *CgiModule*
      iii. Ensure that the **Executable** box is empty.
      iv. In the **Name** box, enter *cgi*, then click **OK**.
6. Grant the following access rights for the Internet Guest account `IUSR` using the **Security** tab on the Windows **Properties** dialog:
   - Full control on `VFHOME\general`
   - Full control on `VFHOME\print`
   - Full control on `VFHOME\webdir\conf`
   - Read-only on `VFHOME\webdir`

The above are the default locations, which can be modified through the ViewDS Management Agent.

7. Test the configuration by typing the Deltawing URL into your browser:

   `http://host:port/ViewDS/Deltawing/webdua.cgi`

   where the port is the port at which the web server is listening (typically 80).

Configuring Apache HTTP Server

To allow Access Presence to connect to the demonstration directory, Deltawing, the following configuration is required for Apache HTTP Server:

- Add a CGI script handler
- Define the document root `${VFHOME}/webdir`
- Add an alias for the Access Presence directory `${VFHOME}/webdir/icons/newicons/`
- Add a script alias for the webdua directory `${VFHOME}/webdir/cgi-bin`
- Ensure there is read access to all directories that comprise the `${VFHOME}` path
- Set up the following access to the directories:
  - all access rights to `${VFHOME}/general`
  - all access rights to `${VFHOME}/print`
  - read-only access to `${VFHOME}/webdir/conf`
  - read-only access to `${VFHOME}/webdir`

`${VFHOME}` is the path to the ViewDS installation directory on Unix. The Windows equivalent is `%VFHOME%` and the default is `C:\Documents and Settings\All Users\Application Data\ViewDS Suite` for Windows 2003 or `C:\ProgramData\ViewDS Suite\` for Windows 2008 onwards.

Example configuration for Windows 2008

To configure an Apache HTTP Server to provide access to Deltawing on Windows 2008:

1. Check that the following line is in Apache `httpd.conf` file:

   ```
   AddHandler cgi-script .cgi
   ```

2. Add the following lines to the end of Apache `httpd.conf` file:

```
DocumentRoot "c:/ProgramData/ViewDS Suite/webdir"
Alias /icons/ "c:/ProgramData/ViewDS Suite/webdir/icons/newicons/"
ScriptAlias /ViewDS/ "c:/ProgramData/ViewDS Suite/webdir/cgi-bin/"
<Directory "c:/ProgramData/ViewDS Suite/webdir">
AllowOverride Options
Options FollowSymLinks
Order deny,allow
Allow from all
</Directory>
```

> This example uses the default ViewDS install location on Windows 2008: `C:/ProgramData/ViewDS Suite`.

3. Make sure that the `webdua.cgi` can write to the directory specified by the `webduaparampath` parameter. The default is `${VFHOME}/webdir/conf`.
4. Restart the web server.
5. Test the configuration by typing the Deltawing URL into your browser:

   `http://[hostname]/ViewDS/webdua.cgi`

Configuring for the XACML framework

This subsection describes the XACML configuration parameters, and includes the steps to modify them through the ViewDS Management Agent. These XACML configuration parameters apply to XACML policy.

- Combining algorithm
- Default version
- RFC822 name attribute
- User base object
- User attributes
- Resource attributes
- Policy base object

Combining algorithm

Access Sentinel can evaluate policies from different sources: native ViewDS XACML policy (defined using the VMA or the Authorization Policy Manager) and non-native XACML policy (either declared in the viewDSXACMLPolicySet attribute or supplied in the request).

When an internal decision request is made only native policies are evaluated. If there is more than one native policy, then the results are always combined using a deny override combining algorithm.

However, when an external decision request is made both native AND non-native policies are evaluated. If a request instructs Access Sentinel to use only policies supplied within that request (CombinePolicy=false), then the evaluation of other policies (for example native policies) will result in a Not Applicable outcome.

If a request instructs Access Sentinel to combine polices supplied within that request and other policies (CombinePolicy=true) then native polices are evaluated using a deny override combining algorithm and non-native policies are evaluated using the combining algorithm specified for that non-native policy set.

The results (native and non-native) are then combined using the Combining Algorithm specified here. Four combining algorithms are available:

- deny overrides – if any nested item (a rule, policy or policy set) evaluates to deny, then the container (a policy or policy set) evaluates to deny; otherwise, if any item evaluates to permit, then the container evaluates to permit; otherwise, the container evaluates to not-applicable.
- permit overrides – if any nested item evaluates to permit, then the container evaluates to permit; otherwise, if any item evaluates to deny, then the container evaluates to deny; otherwise, the container evaluates to not-applicable.
- deny unless permit – if any nested item evaluates to permit, then the container evaluates to permit; otherwise, the container evaluates to deny.
- permit-unless-deny – if any nested item evaluates to deny, then the container evaluates to deny; otherwise, the container evaluates to permit.

For further information see the XACML 3.0 specification.

Default version

Every XACML policy has a version number.

When there are multiple policies or policy sets with the same identifier, the Policy Decision Point (PDP) uses the one with the highest version number. Alternatively, if a Default Version is defined, then the PDP uses the policy or policy set with the highest version number less than or equal to this value.

This parameter only applies to XACML policy that was not defined through the VMA or the Authorization Policy Manager.

RFC822 name attribute

If subject attributes are not provided in an authorization decision request, the Policy Decision Point (PDP) will attempt to look them up in the Policy Information Point (PIP - the ViewDS server). For this to occur the request must include the following XACML attribute:

```
urn:oasis:names:tc:xacml:1.0:subject:subject-id
```

The PDP will look up the subject-id XACML attribute definition from within the XACML Access Control Domain to identify if it has been mapped to a directory attribute. If it has, then the PDP will used this directory attribute to search ViewDS for the subject. If the subject-id does not have a directory attribute mapping, it will use the following defaults based on the subject-id data type:

- String – the Policy Decision Point looks for a directory entry whose viewDSUserName attribute equals the string value specified by subject-id.
- x500Name – the Policy Decision Point looks for a directory entry whose LDA Distinguished Name equals the specified X500 name specified by subject-id
- rfc822Name – the Policy Decision Point looks for a directory entry that has a value of the attribute type identified by the rfc822Name-attribute that is configured within the XACML Configuration setting.

> The PDP only expects to find a single subject entry within ViewDS. If multiple entries are located it will consider the situation to be ambiguous and will not use any of the subject attributes from within the PIP.

User base object

This is the root of the directory subtree in the Policy Information Point (PIP) that the Policy Decision Point (PDP) will search in order to find a user's entry.

User attributes

These are user attributes that the Policy Decision Point (PDP) will need to access when evaluating authorization requests.

Resource attributes

These are resource attributes that the Policy Decision Point (PDP) will need to access when evaluating authorization requests.

Policy base object

The root of the directory subtree that the Policy Decision Point (PDP) will search in order to find a policy or policy set.

Setting the XACML configuration parameters

1. From the ViewDS Management Agent, click **Server View**.
2. In the left pane, click the appropriate **DSA**.
3. In the right pane, click the **XACML Config** tab.
4. Complete the boxes in the **XACML Config** tab as required.
5. At the bottom of the tab, click the **Set XACML Configuration** button.

Exploring ViewDS

This section describes the ViewDS data model to help you become familiar with how ViewDS organizes information. It then takes you through using the demonstration directory, Deltawing, to help you become familiar with ViewDS from a user's perspective. It also provides further experience of using the ViewDS Management Agent.

This section covers the following topics:

- ViewDS data model
- About the demonstration directory
- Exploring through Access Presence
- Exploring through the Management Agent

ViewDS data model

This section describes the following aspects of the hierarchical data model that applies to all ViewDS directories:

- Directory Information Tree
- Context prefix
- Attributes
- Distinguished Name (DN)

Directory Information Tree

A ViewDS directory includes a collection of information about objects in the real world. These objects might be countries, localities, organizations, organizational units, people and job functions, for example.

The objects are represented in the directory by entries arranged in a tree hierarchy. For example, a country contains localities and national organizations, an organization is divided into a hierarchy of organizational units such as divisions and branches, each containing people.

This hierarchical tree is called the Directory Information Tree (DIT).

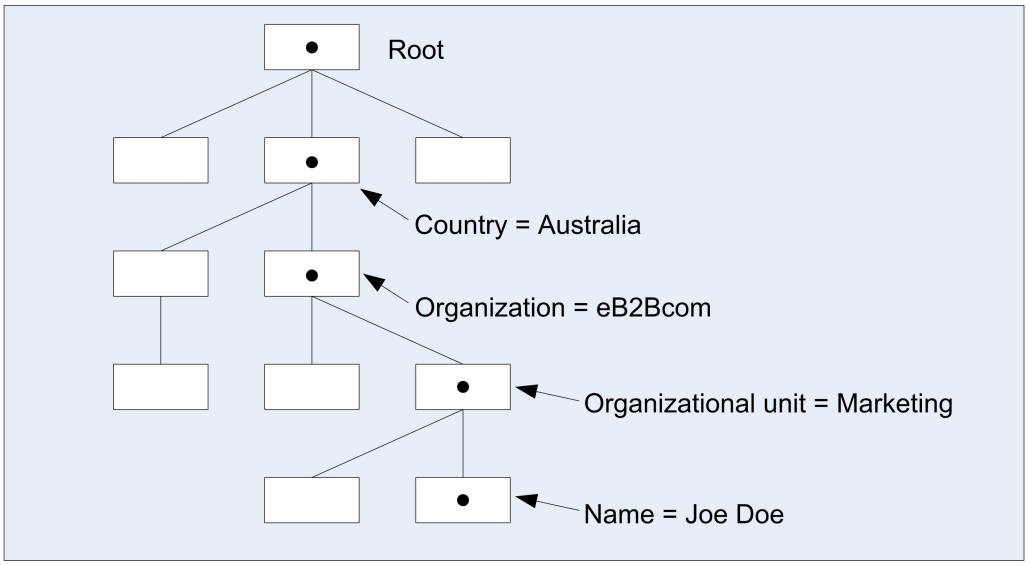


Figure 5: Example DIT

The DIT structure is important when searching a directory because it limits the scope of a search to a specific subtree. The DIT is also the basis for distributing a directory between multiple systems at different sites.

Context prefix

The entry at the top of a discrete area of the DIT is called the context prefix. In the above illustration, for example, there is a context prefix named 'Australia'.

Attributes

Each piece of information held in an entry is called an attribute. An attribute consists of a label identifying the category or type of information (for example, name or telephone number) and a list of one or more values (the actual name or telephone number). This is illustrated below.

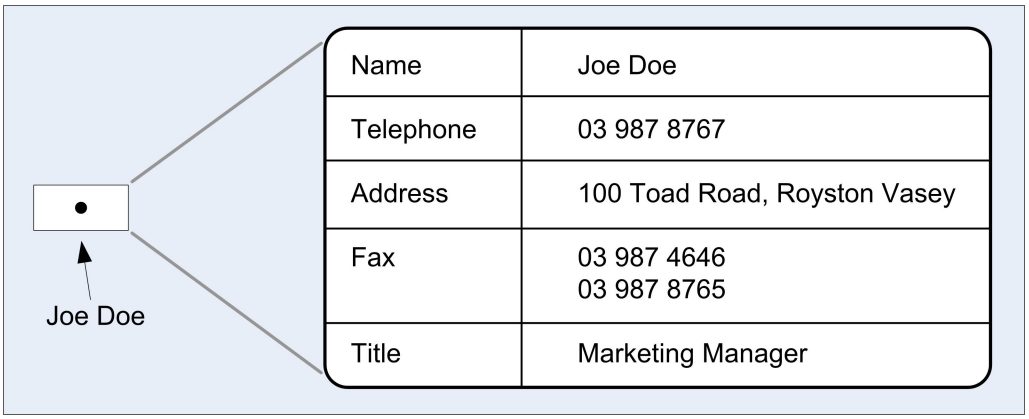


Figure 6: Example attributes

Mandatory and optional attributes

An entry has two sets of attributes: mandatory attributes and optional attributes. An entry is only valid if it has values for its mandatory attributes. The optional attributes without values are not displayed by the DUA.

Distinguished Name (DN)

Every entry has a Relative Distinguished Name (RDN), comprising the values of one or more naming attributes. For example, the naming attribute for the entry in the above illustration might be Name.

The sequence of RDNs from the root of the DIT to an entry forms a unique path, which forms the entry's Distinguished Name (DN).

An entry's DN is formed by adding its RDN to the DN of its immediate superior. For a public white pages directory, for example, the superior might be a locality or a telephone zone. The RDN can consist of multiple components (such as name and address) if this is needed to ensure that all RDNs with the same superior can be uniquely identified.

Aliases

Some entries in the DIT are not objects themselves, but instead, point to the name of an object. These entries are called aliases.

Aliases can be used to make the names of entries more user-friendly, or to provide a transition from an old name to a new name. Aliases also make it possible, for example, for a person who sits on several committees to have one entry that can be found through aliases under each committee.

About the demonstration directory

A new installation of ViewDS includes a demonstration directory called Deltawing. It allows you to explore the functionality of ViewDS from a user’s perspective through Access Presence, and from an administrator’s perspective through the ViewDS Management Agent.

Deltawing is a fictitious, small, and highly diversified company that develops, markets and sells products in the Automotive, Information Systems and Pay TV industries. The directory contains 1014 entries (although a ViewDS directory can store many millions of entries).

Deltawing includes more than just organizational groups and people. The object classes are:

- Units (used for storing organizational departments, branches, sections etc.)
- Organizational Person (used for storing people who work for an organization)
- Working Party (used to store information about work groups such as committees, or teams)
- Role (used to store information about a specific job role in an organization, such as phone, fax, E-mail, and the incumbent person(s) in the role)
- Device (used to store information about physical devices such as printers)
- Meeting Room (used to store information about conference rooms)
- Alias entries (used to store 'pointers' to entries elsewhere in the directory rather than duplicating the entries unnecessarily)

Exploring through Access Presence

This subsection is a tutorial that introduces you to Access Presence in order to give you experience of ViewDS from the user's perspective.

Connecting to Access Presence

To connect to Access Presence:

1. Open the following URL:

   `http://[hostname]/ViewDS/Deltawing/webdua.cgi?`

2. Log on with the user name asherma and password testpass.

Deltawing user names and passwords

The user names, passwords and access levels for the Deltawing directory are as follows:

User name	Password	Access level
rturnbu	testpass	read
cjoyce	testpass	update
asherma	testpass	superuser

Finding your way around

The following activities will help you become familiar with ViewDS from a user's perspective.

Finding and viewing an entry

To search for and view Mike Smith's entry:

1. From the **Welcome** page, click the **Access** button. The **Advanced Search** page is displayed.
2. In the **Surname/Name** box, enter *smith* and then click **Show**. Twelve directory entries are returned that approximately match the name.
3. Click the fifth entry (**Mike Smith**).

   The first section of the page lists the superior entries above Mike Smith in the DIT: deltawing, Deltawing Home Media Ltd, Internet Services, World Wide Web Services, and Strategic Relationships.

   The next section displays Mike Smith's details.

   There are no subordinate entries below Mike – if there were, they would be listed in another section below his details section.
4. Click **Tree Browsing**. The **Deltawing DIT** is displayed with Mike Smith's leaf entry highlighted in bold text.
5. Click **Tree Browsing** again to hide the Deltawing DIT.

Finding the manager of a unit

To find the manager of Delta Home Media Ltd:

6. In the first section of the page, click **Delta Home Media Ltd**. The entry for the unit is displayed.

   The first section of the page shows that this entry has one superior entry, deltawing. The next section contains the unit's details, including a link to its manager Karen Johannesen. The final section contains the unit's subordinate entries.
7. Click **Karen Johannesen**. Her details are displayed.

Searching for entries by function

To find all meeting rooms in Deltawing:

8. At the top of the page, click **Change Search Form**. The **Welcome** page is displayed.
9. In the drop-down box, click **Function Search** and then click **Access**. The **Advanced Search** page is displayed.
10. In the function box, enter meeting and press the return key. A list of meeting rooms is displayed.
11. Click the third meeting room in the list. The entry for the Sales Meeting Room is displayed.

The first section of the page shows that this entry has three superior entries in the DIT. The next section contains the room's details, which includes a link to the person who deals with bookings, Mary Smith.

12. Click **Mary Smith**. Her details are displayed.

> All details are user attributes except for Last Modified, which is an operational attribute.

Modifying an entry

To add a mobile phone number to Mary Smith's entry:

13. Note the date and time for the Last Modified attribute.
14. At the bottom of the page, click **Modify**. The **Modify** page is displayed.

    Note that all attributes are marked for pre-processing, which is defined through the ViewDS Management Agent.

    Also note that Access Controls determine whether a user is allowed to modify an entry.

15. Add a number (any number) to the **Mobile** box and then click the **Save** button. A confirmation dialog box is displayed.
16. Click **OK**. Mary's details page is displayed.

    Note that her entry now includes:

    - under Mobile, the new number you entered
    - under Last Modified, an new date and time
    - under Last Updated, the name of the user you are logged on as

Moving a non-leaf entry

To move Deltawing Automotive's sales department to Deltawing InfoSystems:

17. Click **Tree Browsing**. The DIT for Deltawing is displayed in a new frame.
18. In the DIT, click **Deltawing InfoSystems**. Its details are displayed.
19. In the details frame, click **Set Target Object**.
20. In the DIT below Deltawing Automotive Ltd, click **Sales**.
21. In the details frame, click **Move**. A confirmation dialog box is displayed.
22. Click **OK**. The Sales unit and all its subordinate entries is displayed below Deltawing InfoSystems.

Exploring through the Management Agent

This subsection introduces you to the ViewDS Management Agent, and involves exploring Deltawing through the application. Figure 7: ViewDS Management Agent shows the following main areas of the ViewDS Management Agent interface:

- Menu bar
- Icon bar
- Left pane
- Right pane
- View buttons

The view buttons allow you to move between the following:

- **Server View** - This view allows you to manage one or more ViewDS servers (for example, RAS and DSA status, configuration, logs and replication agreements).
- **Global DIT View** - This view allows you to manage the data stored by a DSA (for example, entries in the Directory Information Tree, schema and access controls).

The content of the left and right panes is different according to whether the Server View or Global DIT View is displayed. (The following example shows the Global DIT View.)

> If your installation is licensed for just ViewDS Access Proxy, only the Server View is available. The Server View and Global DIT View buttons are not displayed.

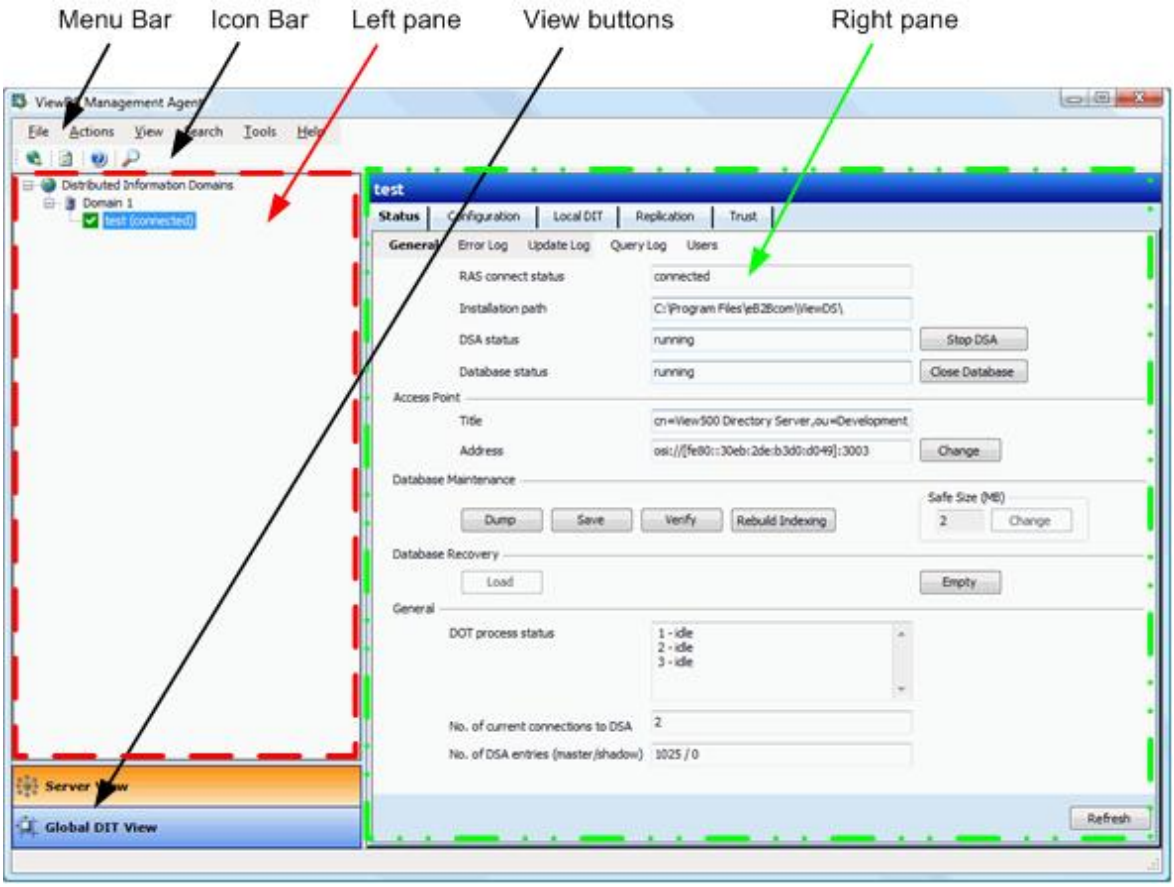


Figure 7: ViewDS Management Agent

Viewing the Deltawing DIT

1. At the bottom of the left pane, click **Global DIT View**. The Deltawing DIT is displayed in the DIT tab on the left.
2. Open and close different branches of the DIT by clicking them.

Viewing an entry's attributes

3. From the **DIT** tab, click the **Deltawing** entry. The Directory Information Tree (DIT) is expanded.
4. Click the entry for **Deltawing Automotive Ltd** and then click the entry for **Tony Liggett**. The **Attributes** tab in the right pane now shows the entry's attributes and their values.

   Note the following:

   - The structural object class for the entry is stored by the attribute objectClass. This entry's structural object class is organizationalPerson (which is a subclass of person).

- The entry's RDN is the value of commonName, Tony Liggett. This value is displayed in the DIT.

Adding an attribute to the entry

5. Right-click anywhere in the right pane and then click **Add Attribute**. The **Add an Attribute** window is displayed.
6. From the **Attribute Type** drop-down box, click **comment**.

   The available attributes for the entry's structural object class are listed by the **Attribute Type** drop-down box.

7. Click **Add Attribute**. The new attribute is added to the **Attributes** tab.

Assigning a value to the new attribute

8. In the **Attributes** tab, right-click the comment attribute and then click **Modify Attribute Value**. The cursor is displayed in the **Value** cell for the attribute.
9. Enter a value and then press the enter key.
10. At the bottom of the right pane, click **Submit**. The changes you have made are saved.

    As comment is a multi-value attribute, you have the option to add another instance of the attribute to the entry by repeating this task from step 5.

Operating the directory

This section describes how to perform operational procedures from the command line and through the ViewDS Management Agent. It covers the following topics:

- Starting and stopping the ViewDS server
- Viewing status
- Creating a new empty database
- Bulk loading a database
- Viewing logs
- Dumping a database
- Backing up a database
- Restoring from a backup

Starting and stopping the ViewDS server

This subsection describes how to start and stop the ViewDS server – Directory System Agent (DSA) and Remote Administration Service (RAS) – and how to open and close its database. The status of the DSA is started or running when its processes are running; its status is terminated when they are not running.

To be available for use, the DSA must be started and its database must be open. Normally, the database is opened when the DSA starts, but some administrative operations on the DSA are only possible when the database is closed. Opening and closing the database are administrative operations on a running DSA.

> All administration tasks require the DSA to be running.

Starting the DSA

You can start the DSA through one of the following:

- ViewDS Management Agent
- Command line
- Windows Services Manager

ViewDS Management Agent

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button. The screen includes the status of the DSA, which is either running or stopped.
4. Click the **Start DSA** button. The DSA Status changes to running.

Command line

1. Log in to the ViewDS host as the system administrator.
2. The RAS is usually configured to start the DSA as part of its own starts-up process. To start the RAS and DSA enter the following command:

   ```
   ras
   ```

Windows Services Manager

1. If ViewDS is not installed as a service, enter the following command:

   ```
   ras -i
   ```

2. Open the **Control Panel** and choose the **Services** option.
3. Select the **ViewDS Administration** service and click **Start**. (The name of the service is set by the configuration-file parameter `rasServiceName`.)

   Click **on Startup** to set other options, such as automatic startup on reboot.

Opening the database

The database normally opens when the DSA starts. If DSA is running but the database is closed, you can open it through either the ViewDS Management Agent or DSA Controller.

ViewDS Management Agent

1. If the **General** screen in the **Status** tab is currently displayed, move to step 5 in this task. Otherwise, continue to step 2.
2. At the bottom of the left pane, click **Server View**. The DSA pane is displayed.
3. In the left pane, click the server.
4. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.

The screen includes the status of the database, which is either closed or running.

5. Click the **Open Database** button. The status of the database changes to running.

Command line

1. Enter the command:

```
dsac open
```

Closing the database

Before you can modify certain configuration parameters, the database must be closed with the DSA running.

ViewDS Management Agent

1. If the **General** screen in the **Status** tab is currently displayed, move to step 5 in this task. Otherwise, continue to step 2.
2. At the bottom of the left pane, click **Server View**.
3. In the left pane, click the server.
4. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button. The screen includes the status of the database, which is either closed or running.
5. Click the **Close Database** button. The status of the database changes to closed.

Command line

1. Enter the command:

```
dsac close
```

Stopping the DSA

The DSA must be stopped before you back up its database or modify certain configuration parameters.

ViewDS Management Agent

1. If the **General** screen in the **Status** tab is currently displayed, move to step 5 in this task. Otherwise, continue to step 2.
2. At the bottom of the left pane, click **Server View**.
3. In the left pane, click the server.
4. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.

The screen includes the status of the database, which is either closed or running.

5. Click the **Stop DSA** button. The status of the database changes to stopped.

Command line

The RAS is usually configured to stop the DSA as part of its own shut-down process.

1. Enter the following command to stop the RAS (and therefore stop the DSA):

```
ras stop
```

Viewing status

This subsection includes procedures for testing the directory's availability, examining the status of the database and dot threads, and listing the current users (the bind list for DUAs and other DSAs).

Checking whether the DSA is running

There are several ways to check whether the DSA is running, which are described below.

ViewDS Management Agent

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.

   The screen includes the status of the DSA, which is either running or stopped.

Start DSA Controller or Stream DUA

A simple way to check whether the DSA is running is to start either the DSA Controller or Stream DUA. Both bind into the DSA when they start up, or print an error message if they cannot connect.

To start the DSA Controller:

```
dsac
```

To start the Stream DUA:

```
sdua
```

Connect through Access Presence

Alternatively, you can try to connect to the directory through Access Presence and see whether you can proceed past the login dialog (or automatic login if applicable). If the directory cannot be accessed, the

following message is displayed:

```
Cannot connect to the directory, try again later
```

Solaris or Linux ps command

You can also use the Solaris and Linux `ps` command (the exact form depends on the operating system). There should be a single `dsa` process and a single `rasrv` process.

Viewing the directory status

The directory's status is the internal state of its processes and the values of its parameters. You can view the directory status through the DSA Status screen of the ViewDS Management Agent or through the DSA Controller's display command.

ViewDS Management Agent

To display the DSA Status screen:

1. At the bottom of the left pane, click the **Server View** button.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button. The General screen for the DSA is displayed.

The General screen includes the following status information:

- **RAS connect status** – the status of the DSA's connection to the RAS. If the DSA is not connected to the RAS, certain information and functions will be unavailable. You will not be able to start or stop the DSA, view logs, or view or modify the DSA's configuration.
- **Installation path** – the installation path for the DSA.
- **DSA status** – the status of the DSA, which is either running or stopped.
- **Database status** – the status of the DSA's database, which is either running or closed.
- **DOT thread status** – the status of each DOT (Directory Operation Thread) thread. The DOT threads provide an interface between the DSA and its database, and also provide ViewDS's flexible searching capabilities.
- **No. of current connections to DSA** – the number of users currently connected to the DSA. The users can include DUAs, LDAP clients, other DSAs and the RAS.
- **No. of DSA entries (master/shadow)** – the number of master entries the DSA contains, and the number of entries it shadows.

Command line

To display the current status and settings, enter the following command:

```
dsac display
```

A typical response is as follows:

```
DSA              : available# line 1
 Database         : open# line 2
 DOT 1            : waiting for an operation# line 3
 Number of dots   : dots       = 1
 Max dot size (MB): dotsize    = 20
 Max sessions     : sessions   = -1
 Max updates      : updates    = 1
 Size limit       : sizelimit  = 2000
 Time limit       : timelimit  = 30000
 Cache size (MB)  : cache      = 2
 DAP timeout      : daptimeout = 0
 DSP timeout      : dsptimeout = 0
 Optimistic       : optimistic = on
 Async mode       : async      = on
 Recovery         : recovery   = on
 Enable ldap      : ldap       = off
 Query logging    : qlog       = on
 Update logging   : ulog       = on
 Activity logging : alog       = off
 LDAP session log : clog       = off
 SEP size factor  : sizefactor = 5
 Search aliases   : searchalias= on
 Safe file initial size(MB)    = 4
```

The first line, DSA, shows that the status of the DSA as one of:

- available
- not available (no DOTs)

- not available (DSA is starting DBM)
- not available (database is closed)
- not available (closing DOTs)
- not available (closing database)

The second line, Database, shows that the status of the database as one of:

- open
- open (being dumped)
- open (being emptied)
- closed

The third line, DOT, shows a status of:

- connecting to DSA
- waiting for an operation
- being closed by DSA
- processing operation
- returning operation
- abandoning operation

If you have set a value for a parameter that applies until the DSA is restarted, then the persistent value is displayed in brackets next to this temporary value. The DSA parameters are described in the ViewDS Technical Reference Guide: Directory System Agent.

Listing the users

The procedures in this section list the users (including DUAs and other DSAs) currently connected to a DSA.

ViewDS Management Agent

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab and then click **Users**. Details about the users connected to the DSA are displayed.

Command line

To list the users currently connected to the DSA, enter the command:

```
dsac userlist
```

The response has the following format:

```
User                      Session              Queries Updates
 ------------------------------------------------------------
 Mr Mark Jones             1997-05-22,16:05:34      14       0
 Mr Bob Chambers           1997-05-22,15:33:31       2       0
 Ms Sue McDonald           1997-05-22,09:02:12      89       9
 Fred Halliday             1997-05-22,15:08:11       7       0
 G.S. Brown                1997-05-22,16:09:58       3       0
 Dr Thomas Castle          1997-05-22,16:15:09      11       0
 Deltawing Test DSA        1997-05-22,15:33:31        177   29
```

It shows the Common Name of the user or remote DSA; the date and time their session started (the time of login); the number of queries requested by the user or DSA; and the number of updates requested by the user or DSA.

Creating a new empty database

Creating a new database involves removing all data and schema, and then creating a new first-level entry (a context prefix) in the database.

ViewDS Management Agent

Emptying a DSA's database

This task empties a directory. It deletes the schema and removes all data from the database. It is advisable to back up the database before performing this task.

To empty a database:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button. The General screen for the DSA is displayed.
4. In the **Database Recovery** area, click **Empty**. All data is removed except for an empty root entry.
5. Follow the steps to add a first-level entry below.

Adding a first-level entry

This task adds a first-level entry to a directory and gives you the option to import a predefined schema and create default X.500 Basic Access Controls. For information about the predefined schema, see the ViewDS Technical Reference Guide: Directory System Agent.

To add a first-level entry:

1. In the right pane, click the **Local DIT** tab. The **Root** entry is displayed in the **Master** tab.
2. Right-click the **Root** entry and then click **Add First Level Entry**. The **Add First Level Entry** window is displayed.
3. Follow the instructions on the screen.

   (Press F1 to view help for the ViewDS Management Agent.)

Command line

Creating a new database from the command line involves starting the DSA, initializing the database's safe file (see page 41) and then emptying the current database.

To empty a database:

1. The RAS is usually configured to start the DSA as part of its starts-up process. To start the RAS (and therefore the DSA) enter the command:

   ```
   ras
   ```

2. Enter the following command:

   ```
   dsac close init open empty
   ```

   You can override the default size for the safe file by including a parameter after the init command, but this is rarely necessary.

You now have a database containing only an empty root entry. You need to add many operational attributes to the root entry and to the first real entries created below the root to set up schema, security, knowledge, DUA configuration information etc. This is normally done automatically when you load or reload bulk data into a database.

Bulk loading a database

This section describes how to load a new or empty database with bulk data. Bulk data is generated when a See " Dumping a database" on page 78, or when data is extracted from a non-ViewDS database and converted.

Bulk data must be in the format of a series of insert (or entry) commands to the Stream DUA (for more information about the Stream DUA, see the ViewDS Technical Reference Guide: Directory System Agent). The entries must be ordered so that a superior entry is inserted before its subordinates.

The following conventions apply to bulk-data files:

- file names should take the form dib.nnnnn
- nnnnn is a sequential number beginning with 00000
- bulk-data files are stored in the dump directory
- each file holds data for approximately 1000 entries

ViewDS Management Agent

Loading a database through the Management Agent involves stopping the DSA. If this is unacceptable, see Stopping ViewDS is unacceptable.

To load a database:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.
4. In the right pane, click the **Stop DSA** button. The DSA status changes to stopped.
5. In the **Database Recovery** area, click the **Load** button. The database may take several minutes to load. When it has loaded, a confirmation window is displayed.
6. Click the **Start DSA** button. The DSA status changes to running.

Command line

There are two options when loading bulk data from the command line:

- If stopping ViewDS is acceptable, use the ViewDS Fast Load (vfload) utility.
- If stopping ViewDS is unacceptable, use Stream DUA.

Stopping ViewDS is acceptable

This option uses the ViewDS Fast Load utility, which offers no protection against fatal errors. If a crash or fatal error occurs you will need to rebuild the database. For this reason, when loading large amounts of data, you may prefer to prepare a load script that breaks up the process to the following steps:

1. Load a number of dib.* files.
2. Copy the ddm.* files to a safe location.
3. Repeat until all files have been loaded.

If you encounter a fatal error, resume the load from the last checkpoint rather than restarting from the beginning.

To stop the DSA and then bulk load a database:

1. Stop the DSA by entering the following command:

   ```
   ras stop
   ```

   > If you want to preserve but disable the existing database files, you can now move the data/ddm.* files to another directory. Never move the data/ddm.* files unless the DSA has been stopped.

2. Run ViewDS Fast Load with the names of the files to be loaded:

   ```
   vfload -dm ../load/*
   ```

   The files are loaded from the load directory.

   > If building of the indexes was deferred by using the empty for filling command, the last command needs to be a fill command. Processing of this fill command may take a long time if the database is very large. This is normal.

3. Start the DSA by entering the following command:

   ```
   ras
   ```

   Users can now access the directory.

Stopping ViewDS is unacceptable

Before loading bulk-data files to a running database, check that none of the files contain an empty command.

There is also a Stream DUA mechanism to consider that helps minimize the impact of the load on users. The Stream DUA's sleep file allows you to specify the time period it should pause between executing operations. For more information, see the ViewDS Technical Reference Guide: Directory System Agent.

To bulk load a database that is running, enter the following command:

```
sdua -dm file1 file2 ...
```

where file1, file2 etc. are the files to be loaded.

Viewing logs

This subsection describes three ViewDS logs:

- Query log
- Update log
- Error log

You can view the logs through a text editor or the ViewDS Management Agent.

Query log

The query log contains users' queries – read, compare, search and list operations – but not the results. The contents of the query log can be replayed using the Stream DUA without any modification (a query that generates an X.500 error is logged, but is commented out with a '#' character and therefore ignored by the Stream DUA).

When query logging is on, all attempted query operations are written to the query log. This log is useful when monitoring performance, tracking problems, or building a file of typical queries.

> The query log is normally off. If left on, the query log file will grow very quickly and the DSA's host will eventually run out of disk space.

ViewDS Management Agent

To turn the query log on or off:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Configuration** tab and then click **Runtime Settings**.
4. For the **Query logging** parameter, select either **on** or **off**.
5. At the bottom of the screen, click the **Set** button.

To view or modify the location of the query log:

1. In the right pane, click **File System**.
2. If required, modify the **Value** for the **Query logging** parameter.
3. At the bottom of the screen, click the **Set** button.

To view the query log:

1. In the right pane, click the **Status** tab.
2. In the **Status** tab, click **Query Log**. The query log is displayed.

Command line

The query logs are written if the DSA operational parameter qlog is on:

```
dsac setwrite qlog = on
```

The query logs are written to the file `qlog` in the directory set by the configuration-file parameter `qlogdir` (by default, `${VFHOME}/logs` or `%VFHOME%\logs`).

Update log

The update log contains all users' update operations – add, remove, modify, move or rename an entry.

The update log is critical to maintaining database integrity after a failure. After restoring a backup, replaying this log will update the database according to all committed transactions since the backup was made.

The contents of the log can be replayed using Stream DUA without any manual modification (an update that generates an X.500 error is logged, but is commented out and therefore ignored by Stream DUA). However, it must first be run through the smerge utility, which ensures that all operations in the log are in chronological order.

ViewDS Management Agent

To view or modify the location of the update log:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Configuration** tab and then click **File System**.
4. If required, modify the **Value** for the **Update logs** parameter.
5. At the bottom of the screen, click the **Set** button.

To view the update log:

1. In the right pane, click the **Status** tab.
2. In the **Status** tab, click **Update Log**. The update log is displayed.

Command line

The update log is written if the DSA operational parameter ulog is on:

```
dsac setwrite ulog = on
```

The update log is written to the file `ulog` in the directory set by the configuration-file parameter `ulogdir` (by default, `${VFHOME}/logs` or `%VFHOME%\logs`).

Error log

The error log records all errors (except X.500 operational errors) encountered by the DSA and co-resident DUA processes. Out of context, individual error messages can be rather cryptic and potentially misleading (for example, warnings and errors may be given when nothing appears to be wrong).

Each log entry has a time-stamp, a process number and error message. The error messages are often only warnings and can be ignored. For example, the log may include messages to say a time limit is too short or there have been too many aborted transactions on a modify command. Neither of these examples would require any intervention.

The error log cannot be switched off and grows indefinitely. However, the standard dbbackup script (Solaris and Linux only) installs an empty error log after completing a weekly backup.

ViewDS Management Agent

To view or modify the location of the error log:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Configuration** tab and then click **File System**.

4. If required, modify the **Value** for the **Error log** parameter.
5. At the bottom of the screen, click the **Set** button.

T**o view the error log**:

1. In the right pane, click the **Status** tab.
2. In the **Status** tab, click **Error Log**. The error log is displayed.

Command line

The error log is written to the file specified by the configuration-file parameter `errorlog` (by default, `${VFHOME}/general/error` or `%VFHOME%\ general\error`).

Dumping a database

When a database is dumped, a text file is generated that contains all data from a specified subtree or from the entire database. This data can be reloaded into the database later.

A dump is a single atomic operation that produces a snapshot of the database. During a dump, the database allows normal operations (including modify) but there can be only one dump process running at a time.

A dump produces a text file (`dib.*`) in the dump directory specified by the configuration-file parameter dumpdir (by default, `${VFHOME}/dump` or `%VFHOME%\dump`). The dump files usually occupy around half the disk space used by the directory database files.

You can dump a database from ViewDS Management Agent or the command line.

ViewDS Management Agent

To dump a DSA's database:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.
4. In the **Database Maintenance** area of the screen, click the **Dump** button.

Command line

There are several options for dumping from the command line:

- Full dump
- Subtree dump
- LDIF dump
- ELDIF dump

Full dump

To dump the entire database:

```
sdua -c dump
```

Subtree dump

To dump a specific subtree:

```
sdua -c dump name
```

Where name is the full Distinguished Name of the entry at the top of the subtree to be dumped. (The name typically extends over several lines, and precise spelling and punctuation are required.)

Alternatively, to avoid potential problems with typing the full Distinguished Name:

1. Use Stream DUA to search for the entry at the top of the subtree to be dumped. For example:

   ```
   sdua -c "search {} for ou ~= \"media\" return" > dumpcmd
   ```

   Where:

   - `media` appears in the name of the entry at the top of the subtree
   - `dumpcmd` is the text file to which the result of the search is written
2. Remove any spurious entries from the text file and replace the keyword entry with dump.
3. Run the following command:

   ```
   sdua dumpcmd
   ```

LDIF dump

To dump the entire database in the LDIF format:

```
sdua -c 'dump as ldif'
```

ELDIF dump

To dump the entire database in the ELDIF format:

```
sdua -c 'dump as eldif'
```

Selective dump

For more control over the entries and attributes dumped, use the Printing DUA described in the ViewDS Technical Reference Guide: User Interfaces.

Backing up a database

A database can be backed up using one of the following commands:

- Save – copies the database files to the save directory (by default, `${VFHOME}/save` or `%VFHOME%\save`). The database remains available for normal operation, including updating, while the command is being processed.

  > It is unsafe to copy the database files in any other way.

- Dump – see Dumping a database.

As well as backing up the database files, you should also back up the See "Command line" on page 76 on a daily basis. The log is critical for maintaining database integrity after a failure. After restoring from a backup, replaying the log will update the database according to all transactions committed since the backup was made.

Daily and weekly backups are recommended.

Daily backup (save)

ViewDS Management Agent

A daily backup involves exporting the update log and then saving the database:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.
4. In the **Database Maintenance** area, click the **Save** button.

   This command saves the database files to the save directory and creates a duplicate of the update log.

   The duplicate has a filename such as `ulog-84b70ab1510`, for example.
5. Use the `smerge` utility to sort the entries in the duplicate update log into chronological order and

write them to a new file with today's date:

```
smerge ulog-84b70ab1510 > ulog.20091021
```

6. Delete the duplicate update log (for example, `ulog-84b70ab1510`).
7. Compress and copy the update log and contents of the save directory to a backup device.

Command line

The steps for a daily backup are incorporated into the `dbbackup` script (see Backup scripts) are as follows:

1. Save the database:

   ```
   sdua -c save
   ```

   This command saves the database files to the save directory and creates a duplicate of the update log.

   The duplicate has a filename such as `ulog-84b70ab1510`, for example.

2. Use the smerge utility to sort the entries in the duplicate update log into chronological order and write them to a new file with today's date:

   ```
   smerge ulog-84b70ab1510 > ulog.20091021
   ```

3. Delete the duplicate update log (for example, ulog-84b70ab1510).
4. Compress and copy the update log and contents of the save directory to a backup device.

Weekly backup (dump)

The weekly backup can be run at any time, but the recommendation is to run it when the load on the directory is minimal.

ViewDS Management Agent

A weekly backup involves exporting the update log and then dumping the database:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.
4. In the **Database Maintenance** area, click the **Dump** button. The dump files are written to the dump directory (by default, `${VFHOME}/dump` or `%VFHOME%\dump`).
5. Compress and copy the contents of the dump directory to a backup device.

Command line

The steps for a weekly backup are incorporated into the `dbbackup` script (see Backup scripts) are as follows:

1. Dump the database:

   ```
   sdua dumpcmd
   ```

   Where `dumpcmd` is a file containing the dump command (see page 1).

   The resulting `dib.*` files are written to the dump directory. They usually occupy around half the disk space used by the DSA's database files.

2. Compress and copy the contents of the dump directory to a backup device (compressing the `dib.*` files typically gives a space reduction of around 10:1).

Backup scripts

Solaris or Linux

Daily and weekly backups can be performed using the dbbackup script, and fully automated using the cron and at commands in a shell script.

Windows

Daily and weekly backups can be automated by incorporating the steps described in the above subsections into a script that runs under the Windows Scheduler.

Restoring from a backup

Restoring save files

You can restore a database from save files using either the ViewDS Management Agent or command line.

ViewDS Management Agent

To restore from save files:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.
4. Click the **Stop DSA** button. The DSA status changes to stopped.

5. Replace the contents of the Database directory (by default, `${VFHOME}/data` or `%VFHOME%\data`) with the saved files on the backup media.
6. Click the **Start DSA** button. The DSA status changes to running.

Command line

To restore from save files:

1. The RAS is usually configured to stop the DSA as part of its own shut-down process. Enter the following command to stop the RAS (and therefore the DSA):

   ```
   ras stop
   ```

2. Replace the contents of the database directory (by default, `${VFHOME}/data` or `%VFHOME%\data`) with the backed-up save files.

3. Start the RAS (and therefore the DSA) by entering the following command:

   ```
   ras
   ```

Restoring dump files

You can restore a database from dump files using either ViewDS Management Agent or the command line.

ViewDS Management Agent

To restore from dump files using ViewDS Management Agent:

1. At the bottom of the left pane, click **Server View**.
2. In the left pane, click the server.
3. In the right pane, click the **Status** tab followed by the **General** tab, and then the **Text View** button.
4. Click the **Stop DSA** button. The DSA status changes to stopped.
5. Replace the contents of the dump directory (by default, `%VFHOME%\dump`) with the backed-up dump files.
6. In the **General** screen, click the **Load** button. This command deletes the current database and replaces it with the contents of the dump files.
7. To update the database according to the transactions that have been committed since the backup was made, follow the steps to See " Replaying the update log" on page 84.

Command line

To restore the database from a weekly backup that uses the dump command:

1. Load the database (see Bulk loading a database).
2. To update the database according to the transactions that have been committed since the backup was made, follow the steps to See " Replaying the update log" on page 84.

Replaying the update log

After restoring a database from a backup, replay the update log to update the database according to all the transactions committed since the backup was made.

To replay the update log:

1. Use the smerge utility to sort the entries in the update log chronologically and write them to a file named with today's date. For example:

   ```
   smerge ulog > ulog.20090327
   ```

2. Copy the sorted update log to the load directory (`$VFHOME/load`).
3. Copy the relevant update logs from the daily backups to the load directory. (These are the log files in backups that were made after the backup you used to restore the database.)
4. Stop the DSA if it is running:

   ```
   ras stop
   ```

5. Run the ViewDS Fast Load utility:

   ```
   vfload -dm ../load/*
   ```

   The content of the update logs are applied to the database.

6. The RAS is usually configured to start the DSA as part of its own starts-up process. To start the RAS (and therefore the DSA) enter the command:

   ```
   ras
   ```

Key concepts – Schema

This section introduces the concepts required to work with ViewDS schema, and describes how to manage schema through the ViewDS Management Agent. It also includes high-level guidance to help you adapt ViewDS to your requirements.

It covers the following topics:

- Understanding schema
- Working with schema
- Understanding search optimization
- Optimizing for searches

Understanding schema

A schema is a set of rules that controls what can be stored in a directory. Every directory has a schema that defines, for example, what kind of entries appear in the Directory Information Tree (DIT), where a particular kind of entry can appear in the DIT and the entry's attributes.

The key concepts for schema are described below.

Subschema area and subschema administrative point

A subschema area is the area of a DIT where a particular set of schema definitions apply. The entry at the top of the subschema area is called the subschema administrative point.

As shown in Figure 8: Subschema area and administrative point, the scope of a subschema area extends to the entries in the subtree below a subschema administrative point.

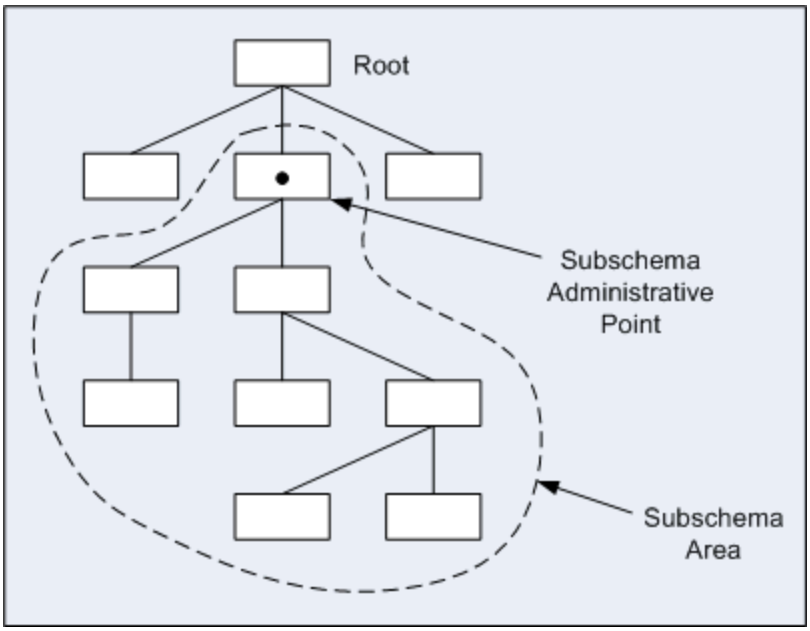


Figure 8: Subschema area and administrative point

A DIT can have as many subschema areas as required, although it is usual to have just one.

Subschema defines objects

Every entry in a DIT has a structural object class defined in a subschema area. Among other things, the definition for a structural object class includes the following:

- Attributes – an object's attributes are either mandatory or optional. A valid entry must have values for its mandatory attributes, but may or may not have values for its optional attributes.
- Content rule – allows you to add attributes to a standard structural object class (in addition those inherited from its superclass). It also allows you to add an auxiliary object class to a standard or non-standard structural object class.
- Name form – defines which attributes appear in an entry's Relative Distinguished Name (RDN). The RDN is displayed in the DIT and in search results.
- Structure rule – defines where entries of the object class can be created in the DIT.

Subschema defines an object's attributes

Attributes are defined independently of the object classes that use them. An attribute definition can be optimized for searches by defining matching rules, approximate matching rules and indexes.

Other key concepts for attributes are described below.

Complex syntax and components

By supporting complex syntax, ViewDS allows you to extend a schema to include complex syntaxes (such as certificates and XML documents) so that their components can be searched. The components are the individual primitive types, which combined constitute a complex syntax.

To illustrate, consider a Human Resources department that stores employees' resumes as XML documents in a ViewDS directory. This would allow people to perform searches on specific areas (components) of the resumes. They might, for example, search the qualifications area to find the employees who have a Masters of Business Administration.

Collective attributes

You can define that an attribute is 'collective', which means it has one value for all its instances in a subtree. This might be useful, for example, for common information such as a departmental fax or phone number.

When a user modifies a collective attribute of an entry, the new value is published to all subordinates of the entry.

Attribute extensions

You can also extend an attribute's properties so that:

- its values are written to a separate file when the database is dumped. This is useful for attributes that store, for example, documents or images as dumped files can be opened with an appropriate application.
- its values are hashed when stored in the database, which is useful for passwords.
- it tracks an entry by its DN.

This last extension is called DN tracking. To illustrate, consider the manager attribute of the Deltawing entry in the demonstration directory Deltawing. By default, this attribute references the DN of the entry for 'Margaret Hunter'. If the entry for 'Margaret Hunter' is moved to another location in the DIT, then DN tracking ensures that the manager attribute in the Deltawing entry still references Margaret’s entry correctly.

Auxiliary object classes

Every entry in a directory is an instance of a structural object class that describes it through a set of attributes. For example, every employee might be an instance of a structural object class called person whose attributes include name, address and phoneNumber.

As well as the structural object class, there are also two other kinds of object class: abstract and auxiliary. The abstract object class is a standard concept in object-oriented theory, and is not discussed here any further.

The auxiliary object class, however, has the following characteristics:

- An instance of an auxiliary object class cannot exist on its own, it must be connected to an instance of a structural object class.
- An auxiliary object class can be used to temporarily tag information to an entry.
- The possible locations of an auxiliary object class are defined by adding it to a structural object class's content rule. (The location of a structural object class is fixed according to the hierarchy defined by its structure rules.)
- An auxiliary object class can be used to identify entries as part of a Basic Access Control policy.

To illustrate how an auxiliary object class might be used, consider the requirement for a directory to have two new kinds of entry: one for junior doctors and one for senior doctors. The information they both need to store includes the usual sorts of things for people, such as email address and phone number, along with information that is specific to either a junior doctor or a senior doctor.

Two ways of fulfilling this requirement are described below.

Using structural object classes

One possible approach is to create two structural object classes, JuniorDoctor and SeniorDoctor, each with a similar set of attributes.

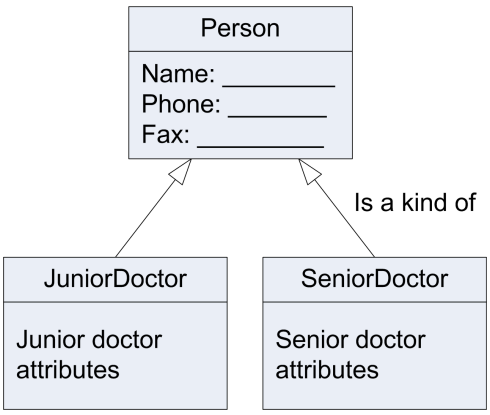


Figure 9: Using structural object classes only

However, because an entry cannot change its class, there would be a problem here when a junior doctor is promoted.

Using auxiliary object classes

An alternative is to create a structural object class, Doctor, and two auxiliary object classes, JuniorDoctor and SeniorDoctor, each with attributes specific to either a junior or senior doctor.

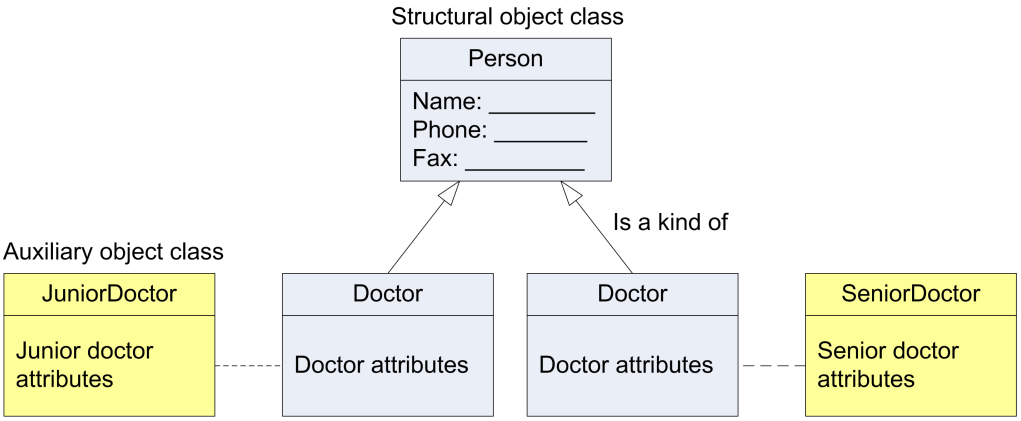


Figure 10: Using auxiliary object classes

The advantage of this approach becomes apparent when a junior doctor is promoted. The only action required is to change their auxiliary class from JuniorDoctor to SeniorDoctor.

Identifying schema elements

Many elements of schema – attributes, object classes, content rules, name forms - have a unique Object Identifier (OID). For further information about OIDs, including how to formulate OIDs for your organization, refer to the ViewDS Technical Reference Guide: Directory System Agent.

Predefined and built-in schema

ViewDS includes predefined schema in text files. You can import one or more of elements of predefined schema – attributes, object classes, content rules, name forms, structure rules, matching rules, etc.

Because an imported definition is part of 'standard' schema, you should not modify it. When the ViewDS Management Agent recognizes a schema definition as being 'standard', you can only modify a subset of its properties such as its description and name. The set of definitions that ViewDS recognizes as ‘standard’ is referred to as built-in schema.

Working with schema

This subsection introduces you managing schema through the ViewDS Management Agent and includes high-level overviews to help you plan and implement a schema according to your requirements.

Viewing schema through the ViewDS Management Agent

1. At the bottom of the left pane, click **Global DIT View**. The DIT tab is displayed.
2. In the DIT tab, click the **Deltawing** entry at the top of the DIT. The following tabs are now displayed in the right pane:
    - **Attributes** – shows the attributes of the Deltawing entry.
    - **Schema** – this tab is displayed because the Deltawing entry is a subschema administrative point.
3. In the right pane, click the **Schema** tab. The Schema tab includes buttons to import and export schema definitions, and contains the following subordinate tabs:
    - **Attributes** – manage the attribute definitions in the subschema area.
    - **Object Classes** – manage object class definitions.
    - **Words** –manage noise words, synonyms and truncated words, which help optimize searches performed by users.
    - **Definitions** – add XML syntax, which can then be used to create an attribute definition.
    - **DUA** – define how different elements of schema are presented by Access Presence.
    - **Indexing** – manage the indexing for attribute definitions.
    - **Matching Rules** – manage the matching rules assigned to attributes in the subschema area.
    - **OID Arcs** – manage Object ID (OID) arcs for attributes, object classes, name forms and matching rules. Defining an OID arc ensures that ViewDS will automatically allocate an OID whenever you create a new schema definition.

Viewing an object class definition

4. Click the **Object Classes** tab. A list of the object classes in the subschema area is displayed.
5. In the **Object Classes** tab, double-click **organizationalPerson**. The organizationalPerson Properties window is displayed.
6. In the window, click the **Attributes** tab. The tab includes two boxes: one for the mandatory attributes, and another for the optional attributes.

7. To view the mandatory attributes, click the **Inherited Mandatory Attributes** button. The Inherited Mandatory Attributes window shows you the attributes that must have values when a new entry of this structural object class is created.
8. Click **OK** to close the Inherited Mandatory Attributes window.
9. Click the **Rules** tab. This tab has three areas: Content Rules, Name Forms and Structure Rules.

Viewing the content rule

10. In the Content Rules box, click **deltawingOrganizationalPerson** and then click the **View** button. The Content Rules window is displayed, which has the following areas:
    - Auxiliary Object Classes – the auxiliary object class specialUser is listed, which means it can be associated with an organizationalPerson entry.
    - Mandatory Attributes – the mandatory attributes in addition to those inherited from organizationalPerson's superclass.
    - Optional Attributes – the optional attributes in addition to those inherited from organizationalPerson's superclass.
    - Precluded Attributes – the attributes that cannot be added to the organizationalPerson's superclass.
11. Click **Cancel** to close the Content Rules window, and then **Cancel** again to close the organizationalPerson Properties window.

Identify schema requirements

Consider the following points when defining requirements for a schema:

1. What entries do you want to store? For example, people, departments, units, etc.
2. What are the sub-categories, if any, for each of the entries? For example, people might divide into contract and permanent staff.
3. What attributes do you want to store in each entry?
4. Should each attribute have a primitive or complex syntax (for example, XML)?
5. Which attribute should be used to uniquely identify each entry?
6. What structure is your data currently stored in?
7. If you intend to use Access Presence, can data in the existing structure be displayed in a browser?
8. How will the directory data be maintained? The structure of the data is important in terms of maintenance because the ViewDS Management Agent allows you to delegate responsibility for maintaining the data in a particular area of the directory. The benefit being that there will be more attention to data management, resulting in more accurate data throughout the directory.

Adapt to your requirements

The following high-level overview will help you adapt a subschema area to your requirements:

1. If the Deltawing subschema matches, or partially matches, your requirements:
    a. Delete the entries in the DIT (see Delete a subtree in the ViewDS Management Agent help).
    b. Go to step 3.
2. If the Deltawing schema does not match your requirements:
    a. Empty the Deltawing directory (see ViewDS Management Agent help).
    b. Identify the predefined schema (see ViewDS Management Agent help) that provides the closest match to your requirements.
    c. Add a first level entry to the empty DIT (see ViewDS Management Agent help). During this task, import the schema you identified in the previous step and create the default Basic Access Controls.
3. Modify the schema according to your requirements. For example, you might decide to do one or more of the following (all of which are described in the Management Agent help):
    - Import additional schema elements that match to your requirements.
    - Add an attribute to a standard object class Modify an attribute.
    - Create an attribute.
    - Modify an object class.
    - Create an object class.
4. Optionally, See " Bulk loading a database" on page 73 into your modified schema.

Understanding search optimization

If you expect an attribute to appear in a user's search criteria, the following allow you to optimize the attribute for searches:

- Matching rules
- Approximate-matching rules
- Word lists
- Indexing

The above can also be assigned to components of an attribute with a complex syntax (for example, an XML document or digital certificate), allowing users to search on the component.

Matching rules

There are three kinds of matching rule – equality, ordering and substrings - which ViewDS uses when searching for an exact match to a user's search criteria.

Equality matching rules

ViewDS uses an attribute's equality matching rule when:

- adding a value to check whether the value already exists
- deleting a value to identify the value to be deleted
- comparing a value using the compare operation

If no equality matching rule is specified, ViewDS cannot distinguish between values of the attribute. A modify operation will still work, but operations that involve matching individual values will not work.

If defined, ViewDS references an attribute's noise words when equality matching.

Ordering matching rules

ViewDS uses an attribute's ordering matching rule when performing a greater-than or less-than search, and when ordering search results.

Substrings matching rules

ViewDS uses an attribute's substrings matching rule when searching for a substring in an attribute's value.

Approximate-matching rules

ViewDS uses approximate-matching rules when a user searches for an approximate match to their search criteria. These rules only apply to attributes, and components of complex attributes, with a string-type syntax.

The approximate-matching rules are:

- Phonetic – for example, the search criteria 'pane' would match the attribute value 'payne'.
- Typing correction – 'Dircetor' would match 'Director'.
- Synonym – 'Bob' would match 'Robert', and 'road' would match 'street' (there is more information about this rule below).
- Prefix – 'thom' would match 'Thomas', 'Thomson' and 'Thomkins' (if defined, ViewDS references an attribute's noise words when prefix matching).
- Suffix – 'son' would match 'Thomson', 'Hodgson' and 'Peterson'.

- Abbreviation – 'NSW' would match 'New South Wales', and 'DOF' would match 'Department of Fisheries' (there is more information about this rule below).

For each of the above, there are two approximate-matching rules:

- keyword matching, which looks at each word in a string individually.
- non-keyword matching, which treats an attribute's value as a discrete string.

To illustrate, for the keyword prefix rule, a search criteria of 'thom' might return 'Thompson', 'Robert Thompson' and 'Thomas Shipman'. For the non-keyword prefix rule, the same search would only return 'Thompson' from the same directory.

There are two more approximate-matching rules, which perform keyword matching only:

- Equality – 'self' would match 'Will Self', 'John Self' and 'Selfish Alfonzo the Third'
- Phonetic Mandarin – performs phonetic matching on Mandarin pronunciation of simplified/traditional Chinese characters

Synonym approximate matching

For synonym approximate matching, you must declare sets of synonyms for each attribute. Otherwise, selecting synonym approximate matching has no effect.

If defined, ViewDS references an attribute's noise words when synonym approximate matching.

Abbreviation approximate matching

When a user's search criteria for an attribute is an abbreviation, ViewDS attempts to match the search criteria against each value of the attribute. ViewDS does this by first removing any noise words defined for the attribute. Then, if a value contains:

- one keyword – ViewDS does not abbreviate it.
- two keywords – ViewDS abbreviates each keyword (automatically or using a truncated word you have defined) and concatenates the abbreviations.
- three keywords – ViewDS abbreviates the value to the first letter of each keyword.

To illustrate, if a user searches on the abbreviation 'DOF', ViewDS might return 'Department of Finance' and 'Department of Fisheries'. If you have declared 'of' as a noise word, however, the abbreviation 'departFish' would also return 'Department of Fisheries'.

The ViewDS Management Agent allows you to define a set of truncated words for an attribute. ViewDS uses these truncated words instead of automatically generating abbreviations when there are two keywords in the attribute's value.

> These methods of abbreviating an attribute's values are also used when an Abbreviated Hierarchy Name is displayed by Access Presence.

Word lists

To improve approximate matching, you can define sets of synonyms, noise words and truncated words for a string-type attribute (or a component of an attribute).

Noise words

A noise word is a word – such as 'the' or 'and' – that is so common that it is usually of little use for searching or indexing. ViewDS ignores the noise words when it performs the following on an attribute:

- indexing
- keyword equality matching
- prefix approximate matching
- abbreviation approximate matching

Declare a set of noise words for an attribute that has any of the above.

Synonyms

The words in a set of synonyms are treated as equivalent when a user requests an approximate match on one of the words in the set (see Synonym approximate matching). For example, a set of synonyms for an attribute might be ‘high school’, ‘secondary college’ and ‘secondary school’. A search on ‘high school’ would return matches on both ‘high school’ and ‘secondary college’.

Truncated words

ViewDS uses an attribute’s truncated words instead of automatically generated abbreviations when there are two keywords in the attribute's value (see Abbreviation approximate matching). They are also used when Access Presence displays Abbreviated Hierarchy Names.

Indexing

Indexing attributes makes searching a directory faster.

An attribute or component that has approximate-matching rules assigned should also have indexing assigned. Fortunately, the ViewDS Management Agent recommends the most appropriate indexes according to the attribute's syntax and matching rules.

If defined, ViewDS references an attribute's noise words when indexing.

Optimizing for searches

The following high-level overview will help you optimize for searches:

1. Identify the attributes that you expect users to search on.
2. For these attributes (see the ViewDS Management Agent help for each task):
    a. assign approximate-matching rules
    b. assign indexing
    c. create a set of noise words
3. For each attribute to which you assigned the synonym approximate-matching rule, create a set of synonyms (see the ViewDS Management Agent help).

Key concepts – Security

This section introduces the concepts required to work with ViewDS security, and describes how to manage security through the ViewDS Management Agent. It also includes high-level guidance to help you adapt ViewDS to your requirements.

It covers the following topics:

- Introduction to security
- ViewDS Access Control
- X.500 Basic Access Control
- Working with X.500 Basic Access Controls
- XACML Access Control
- Working with XACML Access Control

Introduction to security

There are two stages of security when a user accesses the ViewDS directory: authentication and authorization.

Authentication

Authentication involves establishing a user's identity through credentials that are recognized by the Directory System Agent (DSA).

A user can connect (bind) to ViewDS using one of the following levels of authentication: anonymous, simple, and strong.

Anonymous authentication

With anonymous authentication the user connects anonymously without a user name or password.

Simple authentication

With simple authentication the user connects with a DN and password (which is stored in their directory entry). To implement simple authentication each user must be assigned a password (see the ViewDS Management Agent help topic, Implement simple authentication).

Strong authentication

With strong authentication the user is authenticated through a certificate-based mechanism (such as those described by X.509 and TLS).

A user's certificate is stored as a certificate attribute in their DIT entry.

All certificates must be issued by a Certification Authority. The DSA must have a list of the Certification Authorities (CAs) that it trusts regarding the user and CA certificates they issue. The CAs in this list are the 'trust anchors'.

The trust anchors may certify other CAs - intermediate CAs - which may in turn issue certificates or certify further subordinate intermediate CAs. Each intermediate CA that has issued certificates must have a DIT entry that stores its CA certificate.

To manage the list of CAs, see the ViewDS Management Agent help topic Manage Certificate Authorities.

Authorization

After authentication, authorization controls a user's access to the information and services provided by the DSA. Access controls govern which areas of the directory a user is authorized to access.

ViewDS offers a choice of three access-control schemes:

- ViewDS Access Control
  A simple security scheme where each user is assigned one of four levels of access control.
- X.500 Basic Access Control
  A much more versatile option, Basic Access Control allows you to set up any number of Access Control Items (ACIs) to match the different roles of your directory users. It also allows you to group users and assign them the same ACI.
- XACML Access Control
  Allows fine-grained access control that conforms to the XACML Version 3.0 standard.

These three access-control schemes are discussed below.

ViewDS Access Control

This is a simple access-control scheme that applies to a ViewDS directory by default. If a user's entry is not in a domain for Basic Access Control, then ViewDS Access Control applies.

Each user is authenticated anonymously and is then assigned one of four levels of ViewDS Access Control:

- None
  User has no access to the directory, apart from being able to view their own DN.
- Read Access
  User can read all attributes, apart from another user's password, and update their own password. Users have Read Access by default.
- Update
  User can read and update all attributes, apart from privileges and other users' passwords, in a specific subtree.
- Admin Access
  User can read and update all attributes, including all passwords, in a specific subtree.
- Super-user Access
  User can read and update all attributes, including all privileges and passwords, in the entire directory.

If a user does not have one of the above levels individually assigned, the default level applies.

ViewDS Access Control and the ViewDS Management Agent

To implement ViewDS Access Control through the ViewDS Management Agent:

1. Set up anonymous authentication and set the default level of ViewDS Access Control – in the ViewDS Management Agent help, see the topic Configure anonymous privileges.
2. Add the privilege attribute to each user's entry – in the ViewDS Management Agent help, see the topic Configure anonymous privileges.

   If a user's entry does not have the privilege attribute, the default level of ViewDS Access Control applies.

X.500 Basic Access Control

ViewDS implements the X.500 Basic Access Control scheme, and extends it in order to simplify administration and provide greater flexibility. This subsection describes the ViewDS implementation of X.500 Basic Access Control.

Basic Access Control has two main components:

- Access Control Domain

  This is an area of the Directory Information Tree (DIT) containing one or more Access Control.
- Access Control Item (ACI)

  An ACI defines permissions that grant access to entries and attributes in the DIT, and also defines to which users the permissions apply.

In Figure 11: Access Control Domain, there is one Access Control Domain that contains an ACI called 'read only'. It allows all users to search and view the entries in the Deltawing subtree.

Figure 11: Access Control Domain

Access Control Domain

An Access Control Domain is a specific area of a DIT that contains one or more ACIs. Its border is the scope of the ACIs within it.

Usually, just one Access Control Domain is required. An exception, however, is when access needs to be managed autonomously for different areas of a DIT.

The entry at the top of an Access Control Domain is called the administrative point. This is the point from where you can manage the ACIs in the Access Control Domain.

Access Control Item

An Access Control Item (ACI) comprises the following:

- Permissions
- Location
- User class
- Authentication

Permissions

An ACI has a set of permissions that grant access to entries and their attributes. (The permissions can also deny access. However, this should be avoided because users are denied access to the directory by default. Keeping track of ACIs becomes very complicated if some grant access and others deny access to the same areas of the DIT.)

The permissions also have a scope, which is either:

- Individual entries – when the scope of permissions is individual entries, the ACI is termed an Entry ACI.
- Subtrees – when the scope is subtrees, the ACI is termed a Subtree ACI. A Subtree ACI can also have a refinement to identify, for example, all 'people' entries in a subtree.

An Entry ACI should only be used when the location of an ACI cannot be identified by a subtree.

Location

An ACI has one or more locations in the DIT, which is where its permissions apply. An ACI can be located at either:

- Individual entries – an Entry ACI can be located at one or more individual entries; its permissions will apply to each entry.
- Subtrees – a Subtree ACI can be located at the top of one or more subtrees – its permissions will apply to all entries in each subtree. If the Subtree ACI has a refinement, its permissions will apply to the entries identified by the refinement.

By allowing one ACI to have many locations in a DIT, ViewDS simplifies directory management significantly.

User class

A user class is a set of directory users. It defines who the ACI's permissions apply to at each of its locations. The same user class can apply at all locations of an ACI, or a different user class can apply at different locations.

You can specify a user class by selecting one or more of the following:

- individual user entries
- a subtree of users

- a group of users

  A group is an entry of an object class with a multi-valued attribute that holds each group member. The attribute has the syntax DistinguishedName.
- a search filter to identify users

  You can create a role-based ACI by specifying that membership of the user class is determined by information in a user's entry.

An Access Control Domain frequently contains many ACIs with many user classes. When a user belongs to several user classes, ViewDS merges the ACIs that apply to the user. The result might be that several ACIs apply or that some of the ACIs no longer apply to the user.

Authentication

An Access Control also has a minimum level of authentication.

For an Access Control that grants access to an entry, a user can access the entry if:

- they connected to ViewDS with at least the minimum level of authentication; and
- they are in the Access Control's user class for the entry.

For an Access Control that denies access to an entry, a user is denied access if they are in the Access Control's user class for the entry. (Their level of authentication is irrelevant.) However, the same Access Control grants access to a user if:

- they connected to the directory with at least the minimum authentication; and
- they are NOT in any of the Access Control's user classes.

Example 1: One location with one user class

Consider the demonstration directory, Deltawing, supplied with ViewDS:

Figure 12: Deltawing DIT

Now consider a requirement that every user should be able to search and view all entries in the directory.

Fulfilling the requirement

You can fulfil this requirement by creating an Access Control Domain at the top of the DIT, and then creating an ACI at the same location with the following properties:

- Name – the name of the ACI is 'read only' (the name of an ACI should reflect its purpose by saying what it protects or who it applies to).
- Permissions – the ACI grants 'search' and 'read' permissions on the entries in the subtree where it is located. (It is a Subtree ACI.)
- Locations – the ACI has one location, Deltawing, where the permissions apply.
- User class – the user class includes all users in the Deltawing subtree.
- Authentication – the level of authentication required is 'none'.

To illustrate:

Figure 13: Access Control Item (ACI) 'read only'

Example 2: Multiple locations with one user class

In this example, there is the requirement for one person to coordinate the Deltawing meeting rooms. This meeting-room coordinator should be Tony Liggett at Deltawing InfoSystems, and he should be able to 'search', 'read' and 'modify' the following entries:

- Large Conference Room (under the Avalon Factory at Deltawing Automotive)
- Large Meeting Room (under Deltawing Automotive)
- Sales Meeting Room (under Sales at Deltawing Automotive)
- Fishbowl Board Room (under Executive)

Another requirement is that the coordinator must connect to the directory with 'strong' authentication (their identity is confirmed through a security certificate).

Fulfilling the requirement

This requirement can be fulfilled by adding a new ACI to the Access Control Domain created in the first example and by giving it the following properties:

- Name – the name of an ACI should be meaningful and is 'meeting room coordinator' in this example.
- Permissions – the ACI grants 'update' permission on entries where it is located. (It is an Entry ACI.)
- Locations – the ACI has four locations, the meeting rooms, where the permissions apply. (Note that because this is an Entry ACI, any new locations will be individual entries.)
- User class – there is one person in the user class, Tony Liggett.
- Authentication – the level of authentication required is 'strong'. If a user connects with a lower level of authentication, access to the locations will be denied.

> The people in this ACI's user class will also have 'search' and 'read' permissions because the 'read only' ACI also applies to them.

This is illustrated in Figure 14: Access Control Item (ACI) ‘meeting room coordinator’ below.

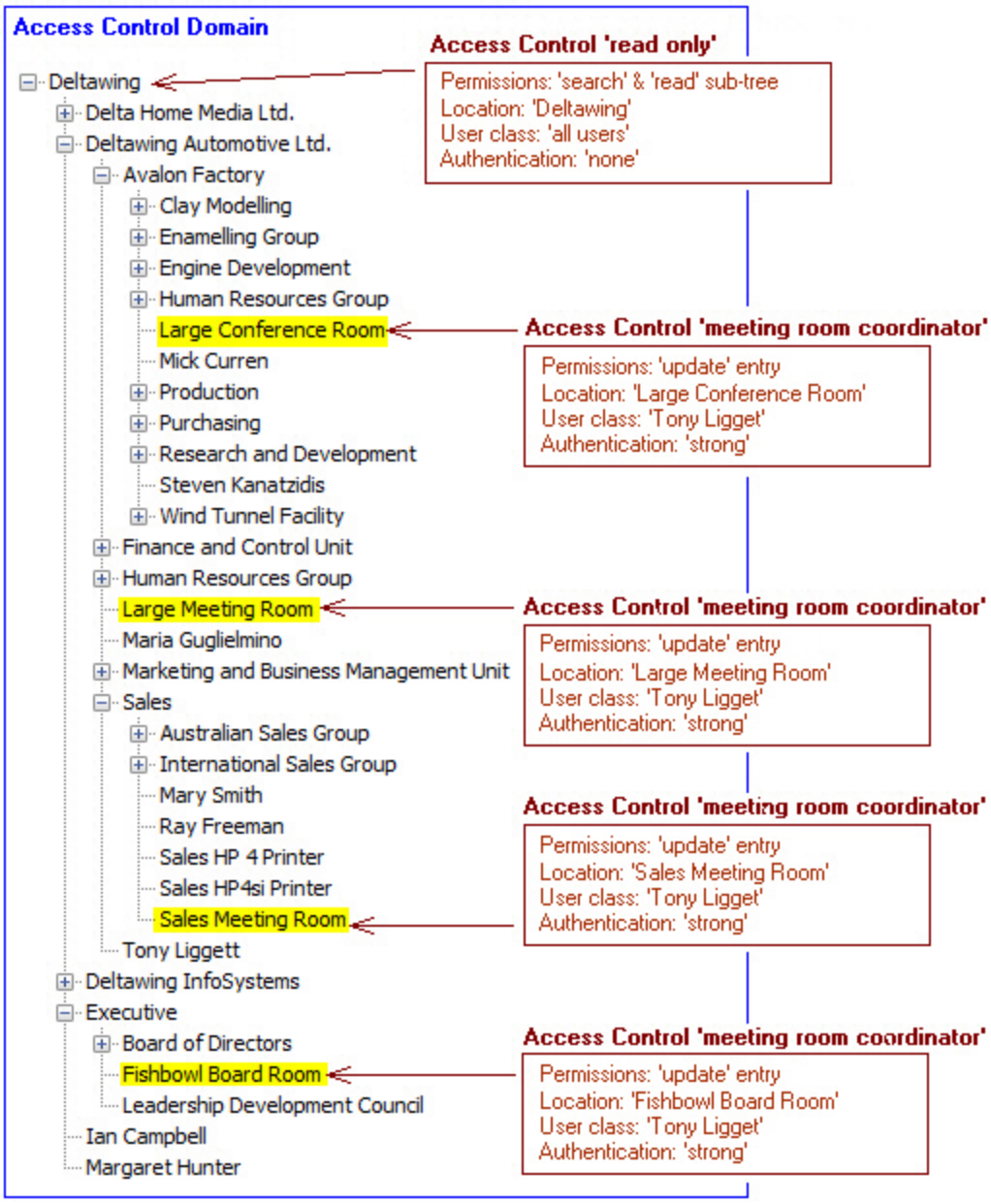


Figure 14: Access Control Item (ACI) 'meeting room coordinator'

Example 3: Multiple locations with multiple user classes

In this example, the requirement is for each division of Deltawing to have its own administrative user who can update users' entries. The following people should have 'search', 'read' and 'modify' access to all users' entries in their division:

- Craig Hunt (Delta Home Media Ltd)
- Maria Guglielmino (Deltawing Automotive Ltd)
- Celine Joyce (Deltawing InfoSystems)

However, they must connect to the directory with 'strong' authentication (their identity is confirmed through a security certificate).

Fulfilling the requirement

This requirement can be fulfilled by adding a new ACI to the Access Control Domain created in the first example:

- Name - the name of an ACI should be meaningful and is 'division admin' in this example.
- Protected items - the ACI grants 'update' permissions on the entries where it is located.
- Locations - the locations are the people entries in each division.
- User class - there is a different user class at each location.
- Authentication - the level of authentication is 'strong'.

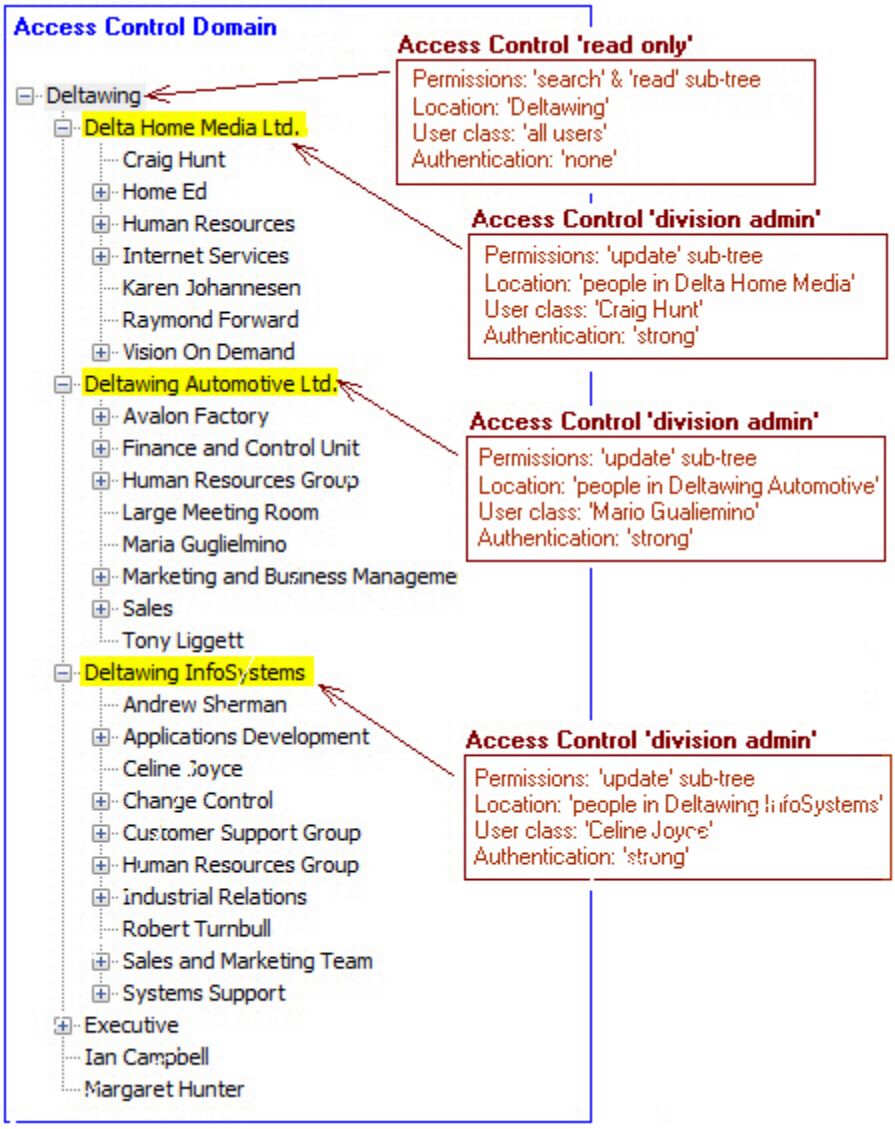


Figure 15: Access Control Item (ACI) 'division admin'

> The people in this ACI's user class will also have 'search' and 'read' permissions because the 'read only' ACI also applies to them.

Example 4: Multiple Access Control Domains

In this example, the requirement is for Deltawing InfoSystems to be completely autonomous. Only employees of Deltawing InfoSystems should be able to search and view this subtree after connecting to the directory using 'simple' authentication.

Fulfilling the requirement

This requirement can be fulfilled by creating a new Access Control Domain at the top of the Deltawing InfoSystems subtree, and then creating a new ACI within it with the following properties:

- Name – the name should be meaningful and is 'InfoSystems read only' in this example.
- Permissions – the ACI grants 'search' and 'read' permissions on the entries in the subtrees where it is located.
- Locations – the ACI has one location, Deltawing InfoSystems, where its permissions apply.
- User class – the user class is everyone in the Deltawing InfoSystems subtree.
- Authentication – the level of authentication required is 'simple'.

The following illustration shows the new Access Control Domain and the one created in the first example:

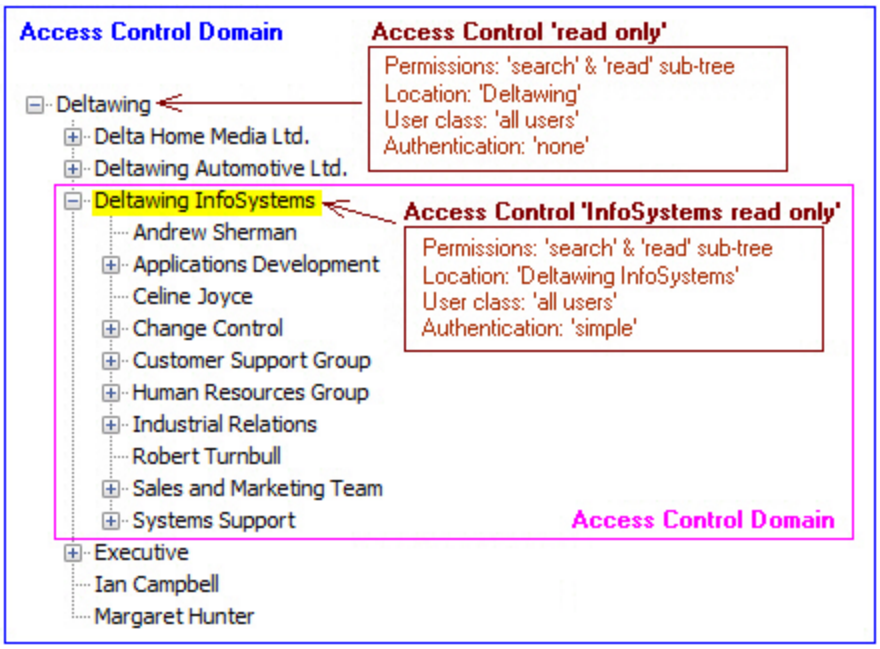


Figure 16: Access Control Item (ACI) 'InfoSystems read only'

An ACI only applies within its Access Control Domain:

- the 'read only' ACI does not apply to users in the Deltawing InfoSystems subtree
- the 'InfoSystems read only' ACI only applies to users in the Deltawing InfoSystems subtree

Example 5: Default Access Controls

Whenever you create a new Access Control Domain, you are presented with the option to automatically create the following default ACIs:

- Own Entry Access
- Read Access
- Super User Access
- Update Access

Own Entry Access

Allows all users to modify their own password, and has the following properties:

- Permissions – grants 'modify' permission on the entry and password attribute where it is applied. (Granting permission to modify an attribute involves granting 'modify' twice: first, on the entry permission, and then on the specific attribute.)
- Location – all entries in the Access Control Domain.
- User class – the user class is 'this entry', which refers to the user who connects to the directory with the same RDN as the entry being accessed.
- Authentication – simple.

Read Access

Allows all users to search and view the entries in the directory, and has the following properties:

- Permissions – grants 'search' and 'read' permissions on the entries where it is located.
- Location – all entries in the Access Control Domain.
- User class – all users in the Access Control Domain.
- Authentication – 'simple'.

Super User Access

Provides full access to all entries in the directory, and has the following properties:

- Permissions – grants all permissions on all entries and attributes, including user names, passwords and some operational attributes.
- Location – all entries in the Access Control Domain.
- User class – none assigned.
- Authentication – 'simple'.

Update Access

Allows a user to update entries in the directory, and has the following properties:

- Permissions – grants all permissions (except 'search' and 'read') on all entries and attributes (except users' passwords). It also grants access to several operational attributes.
- Location – all entries in the Access Control Domain.
- User class – none assigned.
- Authentication – 'simple'.

Working with X.500 Basic Access Controls

This subsection introduces you managing X.500 Basic Access Controls through the ViewDS Management Agent. It also includes a high-level overview to help you plan Basic Access Controls according to your requirements.

Basic Access Control and the ViewDS Management Agent

To add the default Basic Access Controls to Deltawing:

1. If the DIT tab is not displayed, click **Global DIT View** at the bottom of the left pane.
2. Click the **Deltawing** entry at the top of the DIT. The entry's attributes are displayed in the Attributes tab.
3. Right-click the **Deltawing** entry and then click **Add Access Control Domain**. A confirmation window is displayed.
4. In the confirmation window, select the check-box and then click **OK**. The ACI tab is added to the right pane.
5. Click the **ACI** tab.

   The default ACIs are listed in the Subtree ACI tab: Own Entry Access, Read Access and Superuser Access. A fourth default ACI, Update Access, is listed in the Entry ACI tab.
6. Double-click the first ACI in the list, **Own Entry Access**.

   The Own Entry Access window is displayed and includes the following information about the ACI:

   - The name, precedence, authentication level and scope. The scope is Subtree, which tells you that when the ACI is located at an entry in the DIT, its permissions apply to all the entry's subordinates.
   - In the Entry Permissions area, the modify permission is granted. This tells you that the users in the ACI's user class can modify the entries where the ACI is located.

- In the User Attribute Permissions area, modify access is granted for the userPassword attribute. This tells you that the users in the ACI's user class can modify this attribute for the entries where the ACI is located.
- In the Operational Attribute Permissions area, modify access is granted for the userConfig operational attribute. This tells you that the users in the ACI's user class can modify this attribute for the entries where the ACI is located.
- The box on the left of the window, Locations of ACI, shows the locations of the ACI. The entries where the ACI is located are displayed in pink text. You can see that this ACI has one location, Deltawing. As its scope is 'subtree', this means that the ACI's permissions apply to the Deltawing entry and all its subordinate entries.

7. To view the user class for the ACI, right-click the **Deltawing** entry and then click **Edit User Classes**.

   The User Classes window is displayed. The 'This entry' box is selected in the window, which tells you that the user class comprises every user who connects to the directory with the same RDN as the entry being accessed. The result is that each user can modify their own entry.

8. Click **Cancel** to close the User Classes window, and then click **Cancel** again to close the Own Entry Access window.

Identifying requirements

Consider the following for a new installation:

1. Do you need more than one Access Control Domain? (Usually, just one Access Control Domain is required. An exception, however, is when access needs to be managed autonomously in different areas of a DIT.)
2. What are the different roles and groups of users that need access to the directory:
   - What are the groups of users that need different levels of access?
   - What is the basic level of access you want to allow all users?
   - How will you identify these user groups (as individual entries, a group of entries, a role identified by a refinement)?
   - What special access do you want to grant to a user to their own entry (for example, 'self service')?
   - Are any of your requirements fulfilled by the default See "Example 5: Default Access Controls" on page 108)?

For each role or user group you have identified:

1. Which locations in the directory should they have access to?
2. Can these locations be identified as entire subtrees, refinements of subtrees, or individual entries? (The last option, individual entries, should only be used when the locations cannot be identified as subtrees or refinements. Another alternative is to use an auxiliary object class to identify the entries to which the ACI should apply.)
3. Which operational and user attributes at each location do you want to protect?
4. Should the user class be the same at each location of the ACI?
5. What are the permissions in the existing ACIs that apply to these users?
6. What permissions should apply to the protected entries and attributes? (All access is denied by default, and it is usually unnecessary to use an ACI to deny access. Access should be granted on an 'as needed' basis.)
7. What level of authentication is required by the users?

For the steps to create an Access Control Domain and to create ACIs, see the ViewDS Management Agent help.

XACML Access Control

This section provides the background information required to apply the XACML Access Control scheme to a directory by writing XACML policy.

Brief introduction to XACML

XACML Version 3.0 is a standard that provides a framework for fine-grained access control. The standard describes two languages, both written in XML: an access-control policy language; and an access-control decision language.

The policy language is used to specify access-control requirements by defining policies that describe, for example, who can access what and when. The decision language is used to form requests and responses. A request asks whether a given action by a given entity should be allowed; and a response provides the answer, which is determined according to an XACML policy.

Simplified XACML implementation

Figure 17: Attempt to access resource illustrates a simplified XACML implementation.

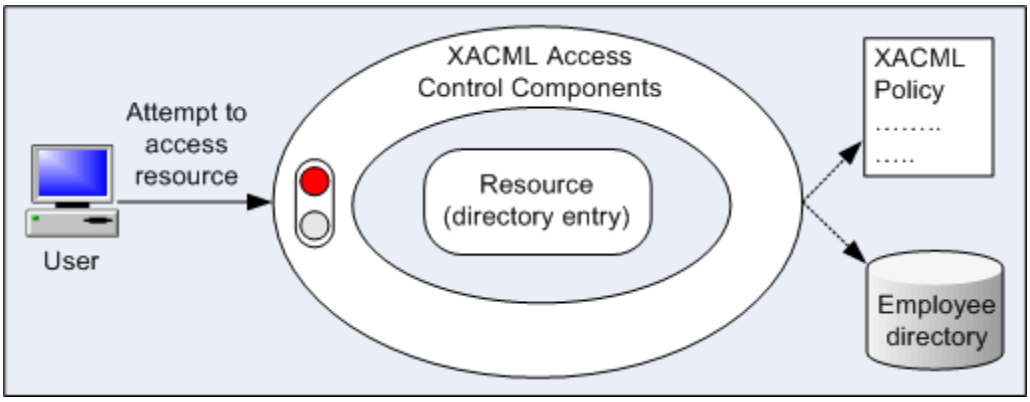


Figure 17: Attempt to access resource

In Figure 17: Attempt to access resource, a user attempts to view a directory entry protected by an XACML access-control implementation. The implementation determines whether the user should be permitted or denied access by interrogating the appropriate XACML policy.

The policy might include considerations such as the user's security level, department, role, position, location and the time of day. All combine to determine whether the user should be allowed access to the resource (as shown below).

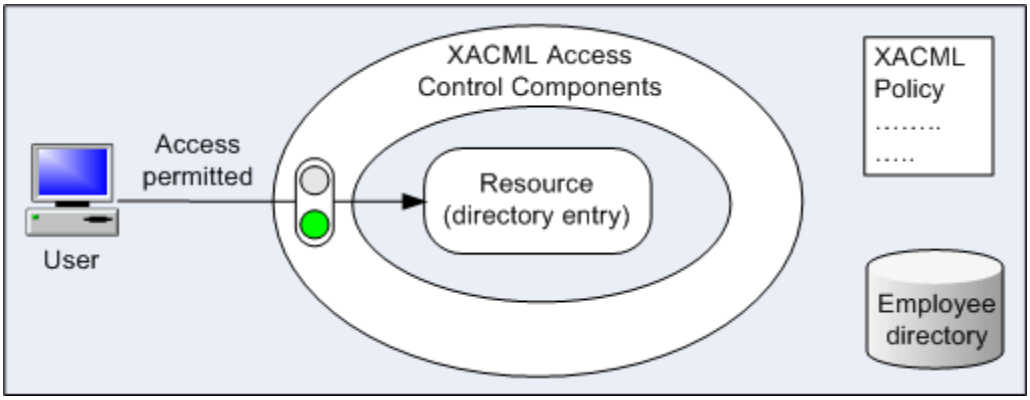


Figure 18: Permit access to resource

Components of the XACML framework

Figure 19: ViewDS XACML framework shows the logical components of the ViewDS XACML framework.

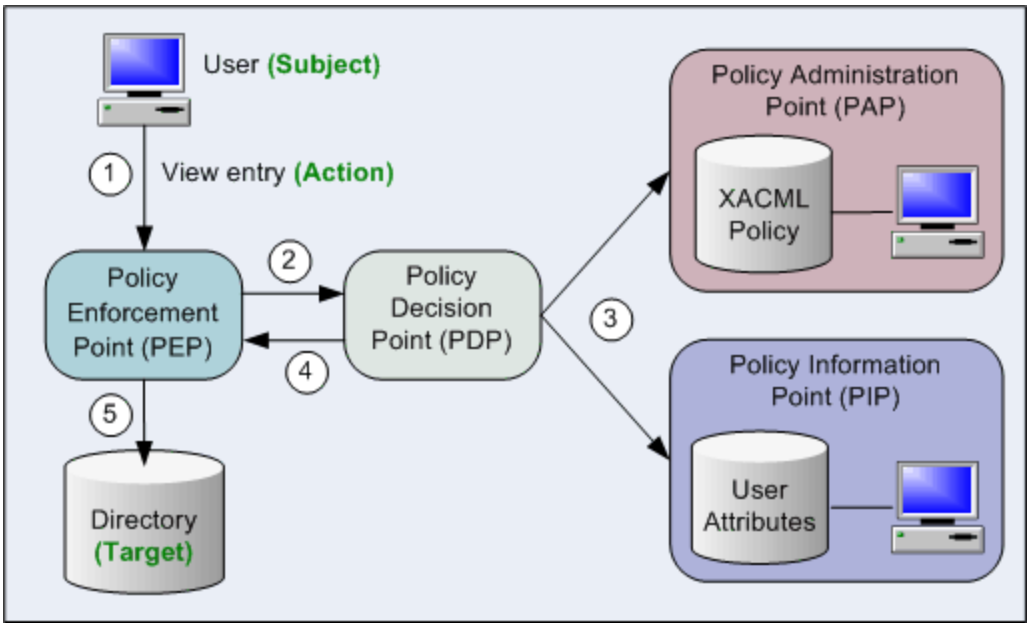


Figure 19: ViewDS XACML framework

The PDP and PEP are within the ViewDS Directory System Agent (DSA). The three data sets in Figure 19: ViewDS XACML framework are shown as separate components for the sake of clarity. The PAP's XACML policies, the PIP's user attributes, and the directory are all stored in the ViewDS database and can be managed through the ViewDS Management Agent.

The steps shown in Figure 19: ViewDS XACML framework are as follows:

1. A user attempts to view an entry in the ViewDS directory.
2. The PEP sends an 'authorization decision request' to the PDP. The request includes XACML attributes that identify (among other things) the user, the entry they are attempting to access, and the action they are attempting to perform. (See the ViewDS Management Agent help topic, XACML attributes provided by the ViewDS PEP.)
3. The PDP determines whether access should be permitted. It looks at the appropriate XACML policy in the PAP, and the appropriate user attributes in the PIP (the user may be identified according to directory attributes in the PIP).
4. The PDP returns an 'authorization decision response' to the PEP, which enables ViewDS to act on the decision to permit or deny access to the document.
5. If access has been permitted, the user is allowed to view the entry.

XACML terms to remember

There are a couple of important XACML terms to remember:

- Target – the set of resources protected by the policy.
- Resource – the specific item (e.g. web page) within the target that the subject is attempting to access.
- Subject – the user attempting to access a resource.
- Action – the action attempted by the subject (e.g. view a web page).

These terms are illustrated below:

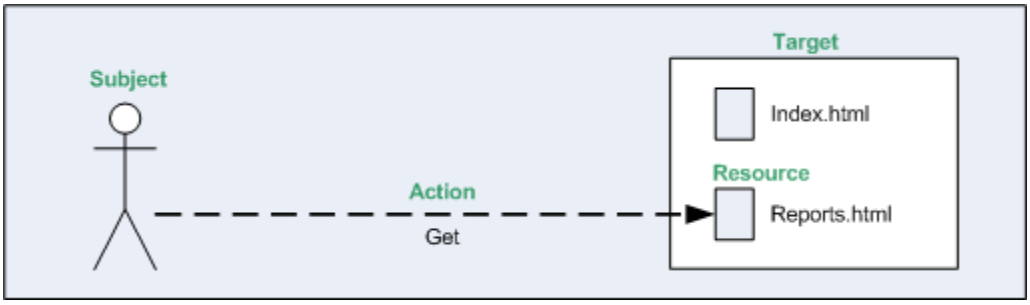


Figure 9: XACML terminology

XACML policy components

The Access Sentinel implementation of an XACML policy comprises:

- XACML Access Control Domain
- Status and version
- XACML attributes
- Rules

Each is discussed below.

XACML Access Control Domain

An XACML Access Control Domain is a specific area of a DIT that contains one or more XACML policies.

> In the ViewDS implementation of XACML, the default behaviour is to deny access to the entities within an Access Control Domain. (This does not apply to administrative users of the ViewDS Management Agent, who bypass all access controls.)

For example, when working with the ViewDS directory and the internal PEP, an XACML Access Control Domain is an area of the directory where the XACML access controls apply. The entry at the top of the

domain is termed the access control administrative point. By default, Access Sentinel denies access to all entries within the domain.

Status and version

Every XACML policy has a status and version.

A policy can have multiple versions, each with a unique version number. A version also has a status that identifies whether it is 'locked' and 'active'.

Only one version of a policy can be 'active'. This is the version that currently applies. You can therefore test a new version of a policy and then roll-back to a previous version if necessary.

A 'locked' version cannot be modified. However, you can create a new version based on an existing locked version. This offers a level of version control.

XACML attributes

XACML is based on the concept of attributes.

The PAP uses XACML attributes to identify the subject, resource, action and environment information within a rule. The PEP sends requests made up of XACML attributes to the PDP to convey information about the subject, resource, action and environment. The PDP then compares these to attribute values in a policy to make access decisions.

The XACML standard defines four categories for attributes:

- Subject – which identify the subject attempting to access a particular resource.
- Resource – which identify the resource the subject is attempting to access.
- Action – which identify the action the subject is attempting to perform on the resource (for example, read, modify).
- Environment – which identify environmental factors such as the day of the week and time of day.

> It is permissible within the XACML standard for any of these four categories to be sub-divided or for other new attribute categories to be added.

For details of the XACML categories and data types of the attributes provided by the ViewDS PEPs, see XACML attributes provided by a PEP.

For an XACML attribute to be included in policy rules, it must first be declared in the XACML Access Control Domain. Declaring an XACML attribute involves giving it a 'user-friendly' name. This is important

because XACML attributes are identified by long URIs or complex XPath expressions that are unwieldy when creating rules.

Access Sentinel allows you to declare two different types of attributes: attribute designators and attribute selectors.

Attribute Designators

An attribute designator comprises the Category, AttributeId and DataType URIs of a particular XACML attribute.

For some XACML attributes, the declaration also includes a mapping to a directory attribute in an entry that uniquely identifies a subject or resource.

Attribute designators allow a policy to specify an attribute value with a given category, identifier and data type. The PDP will then look for that value in the request, or elsewhere, if no matching values can be found in the request (see Attribute look-up).

Attribute Selectors

In addition to XACML attributes, XACML requests can contain XML documents for each category. For example, an XML document might describe the subject or be the actual resource being accessed.

Attribute selectors allow a policy to look for attribute values in such XML documents using XPath queries.

XPath is a language, based on a tree representation of XML documents, which provides the ability to navigate around the tree and select nodes using a variety of criteria.

An attribute selector comprises a category, data type and an XPath expression. Together these are used to resolve a set of attribute values in the request document.

Attribute selectors can be used within XACML policy expressions in the same way as attribute descriptors. For example, consider an XACML request that contains an XML document which is the resource a user is attempting to access. An attribute selector can be configured with an XPath expression to find elements in the document named PublicationDate. An XACML policy can then include a condition that denies access if the PublicationDate is more than five years ago.

Access Sentinel currently supports:

- the definition of attribute selectors within the Authorization Policy Manager (and the ViewDS Management Agent)
- the ability to use and evaluate attribute selectors within XACML policies

However, attribute selectors are not supported by the following as they do not make use of XML documents within authorization decision requests:

- the ViewDS XACML framework
- the HTTP PEPS (IIS and Apache)

Rules

A rule allows the Policy Decision Point (PDP) to determine whether a subject should be permitted or denied access to a resource. Each has a target, scope, an effect (permit or deny access) and a condition.

The target identifies the resources protected by the policy. The scope is used when defining policy for hierarchical resources, such as directory entries. It determines whether the policy applies to a single target resource (entry), or to a target resource and all its subordinates (subtree).

The condition incorporates XACML attributes which the PDP uses to identify the resource and subject. It determines whether the rule's effect should be applied.

A simple example rule is shown below.

Rule:
  Target: Documents
  Scope: subtree
  Effect: Permit access (if the condition is true)
  Condition:
resource has attribute webpage = 'index.html' AND
subject has attribute role = 'Board Member' AND
action = READ

The condition is true if the subject is a Board Member attempting to view the resource 'index.html'.

Working with XACML Access Control

This tutorial takes you through the steps required to write and apply an XACML policy to an area of the demonstration directory provided with ViewDS, Deltawing. Before starting the tutorial read XACML Access Control.

The tutorial includes the following:

Overview

This tutorial's requirement is for a policy that gives one user, Andrew Sherman, the privileges to modify meeting room entries in the Deltawing directory.

Both Andrew Sherman and a meeting room can be identified in the Deltawing directory by their entries' directory attributes. Andrew can be identified by his entry's viewDSUserName attribute which is set to 'asherma'. And a meeting room entry can be identified by its businessCategory attribute which is set to 'Meeting Room'. (If inclined, you can search for these entries through the ViewDS Management Agent – see the Help topic Search the DIT.)

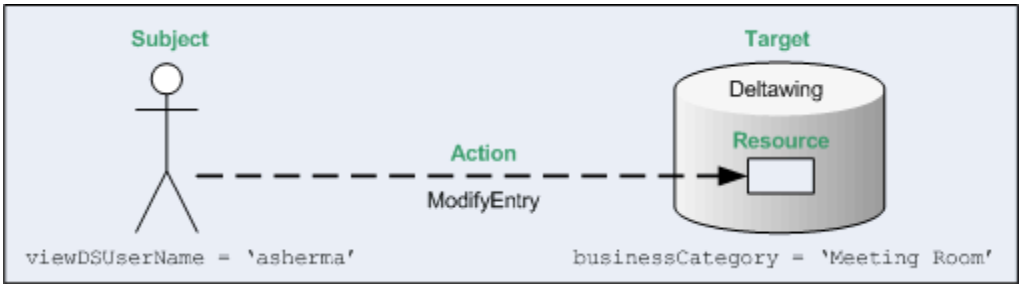


Figure 21: Policy requirement

When a directory user (subject) attempts to modify an entry (resource), the Policy Enforcement Point (PEP) will send an authorization decision request to the Policy Decision Point. The request includes the values of directory attributes in the subject and resource entries, plus a value to identify the attempted action. These values are held in XACML attributes.

XACML attributes

Before an XACML attribute can be used by the PAP, it must first be declared in the XACML Access Control Domain.

Each declaration has a 'user friendly name' that will appear in a rule's condition, an XACML category, and may also require a mapping to a directory attribute.

In this tutorial, the following declarations are required.

User Friendly Name	XACML Attribute Category	XACML Attribute Identifier	XACML Data Type
User Name	urn:oasis:names:tc:xacml:1.0:subject-category:access-subject	urn:oasis:names:tc:xacml:1.0:subject:subject-id	string
Action	urn:oasis:names:tc:xacml:3.0:attribute-category:action	urn:oasis:names:tc:xacml:1.0:action:action-id	string
Business Category	urn:oasis:names:tc:xacml:3.0:attribute-category: resource	businessCategory (urn:oid:2.5.4.15)	string

An XACML attribute's category corresponds to its purpose, as illustrated in Figure 21: Policy requirement.

Business Category is mapped to the directory attribute businessCategory through its XACML Attribute Identifier. However, User Name does not need to be mapped to a directory attribute because it is one of three values the PEP provides to identify the subject.

For details of the XACML attributes provided by the PEP see XACML attributes provided by the ViewDS PEP.

Rules

Two rules are required. The first will permit Andrew Sherman to modify meeting room entries in the directory. The second will permit all users to search and view entries in the directory.

The second rule is required because the default behaviour is to deny access within an Access Control Domain, unless explicitly permitted.

The first rule's target, scope, effect and condition are shown below.

Rule 1:
Target:Deltawing
Scope:Subtree
Effect: Permit access (if the condition is true)
Condition:resource has attribute Business Category = 'Meeting Room' AND
subject has attribute User Name = 'asherma' AND
(Action = 'ModifyEntry' OR Action = 'AddType OR
Action = 'RemoveType' OR Action = 'AddValue' OR
Action = 'RemoveValue')

The rule's target will be the entry at the root of the Deltawing directory, and its scope will be the entire subordinate subtree below the root entry.

Its effect will be to permit access if the condition is true. The condition will be true when the user with the User Name 'asherma' (subject) attempts one of the actions on a meeting room entry (resource). Note that omitting the resource clause would make the rule more general so that it applied it to all entries in the directory.

The second rule's target, scope, effect and condition are shown below.

Rule 2:
Target:Deltawing
Scope:Subtree
Effect: Permit access (if the condition is true)
Condition:Action = 'ReadEntry' OR Action = 'BrowseEntry' OR
Action = 'ReturnDN' OR Action = 'ReadType' OR
Action = 'FilterMatchType' OR Action = 'ReadValue' OR
Action = 'FilterMatchValue'

It has the same target and scope as the first rule. It also permits access if the condition is true. The condition will be true when any user (subject) attempts one of the search or read actions on any directory entry (resource).

Create an XACML Access Control Domain

An XACML Access Control Domain is a specific area of a DIT that contains one or more XACML policies. The entry at the top of the domain is termed the access control administrative point.

To create an XACML Access Control Domain:

1. In the ViewDS Management Agent, click **Server View**.
2. In the left pane, click your ViewDS server. The Status tab displays the status of your ViewDS server. Ensure that the ViewDS Management Agent is connected to your ViewDS server, and that your ViewDS server is running.
3. In the bottom left pane, click **Global DIT View**.
4. Press **F5** to refresh the screen.
5. In the left pane, expand the **Deltawing** entry at the top of the Directory Information Tree (DIT).
6. Right-click the **Deltawing** entry. A submenu is displayed.
7. From the submenu, click **Add XACML Access Control Domain**. The XACML AC tab is added to the right pane.

Declare XACML attributes

To declare the XACML attributes for the tutorial's policy:

1. In the right pane, click the **XACML AC** tab.
2. Within the XACML AC tab, click the **Attributes** tab.
3. At the bottom of the right pane, click the **New button**. The XACML Attribute window is displayed.
4. In the **Label** box, enter *Action*.
5. In the **Category** box, click **urn:oasis:names:tc:xacml:3.0: attribute-category:action**. The Identifier box defaults to urn:oasis:names:tc:xacml:1.0: action:action-id, and the Data Type box defaults to string.
6. In the Permitted Values area, click the **Add** button. A dialog is displayed.
7. Enter *ReadEntry* and click **OK**. The value is added to the Permitted Values box.
8. Repeat steps 6 and 7 to define the following as permitted values: ModifyEntry, ReadEntry, BrowseEntry, RemoveType, AddType, AddValue, RemoveValue, ReturnDN, ReadType, FilterMatchType, ReadValue, FilterMatchValue, DiscloseValueOnError, DiscloseTypeOnError, DiscloseEntryOnError.
9. Click **Save**. The XACML attribute is added to the Attributes tab.
10. Repeat steps 3 to 5 to declare the following XACML attributes.

The XACML attribute Business Category is mapped to the directory attribute businessCategory.

User Friendly Name	XACML Attribute Category	XACML Attribute Identifier	XACML Data Type
User Name	urn:oasis:names:tc:xacml:1.0: subject-category:access-subject	urn:oasis:names:tc:xacml:1.0: subject:subject-id	string
Business Category	urn:oasis:names:tc:xacml:3.0:attribute-category:resource	businessCategory (urn:oid:2.5.4.15)	string

Create a policy

To create the policy:

1. In the XACML AC tab, click **Policy Versions**.
2. In the right pane, click **Version Management** button followed by **New Policy Version**. The XACML Policy Version window is displayed.
3. Accept the default values by clicking **Save**. The new policy version number and its status is displayed next to the Version Management button.

The policy is marked as open, which indicates that it can be modified. Once a policy has been locked it cannot be modified. You can, however, create a new policy based on it.

Define the first rule

To define the first rule:

1. With **ABAC Rules** and **Access** selected in the filter boxes, click the **New** button. The XACML Rule window is displayed. It allows you to define a rule for the current policy.
2. In the **Label** box, enter *Andrew Sherman full access to meeting rooms*.
3. In the **Description** box, enter a short description of the rule, such as *Permit Andrew Sherman full access to all meeting room entries*.

The Target is set to Deltawing and its Scope is subtree. Hence, the target is all subtrees and entries subordinate to Deltawing. Also note that the Effect is set the default, permit.

4. Click the **Edit** button. The XACML Expression window is displayed, and is described below.

XACML Expression window

Figure 22: XACML Expression window shows the XACML Expression window, which allows you to define the expressions that constitute a rule's condition. The window has the following areas:

- Expression Tree – the window's main work area and allows you to build the expressions in a rule's condition in a tree format.
- Text Pane – shows the contents of the Expression Tree in a plain text format.

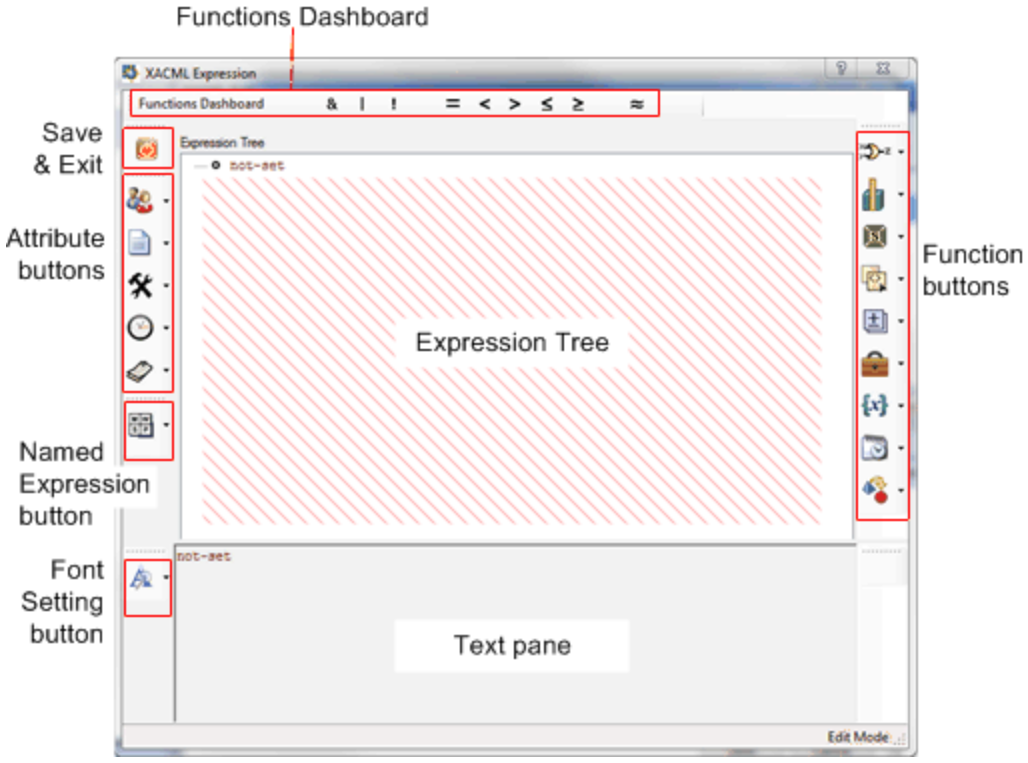


Figure 22: XACML Expression window

The window also has the following buttons:

- Functions Dashboard – allows you to add one of the frequently used functions to the Expression Tree. The functions are also available through the function buttons.

- Save and Exit button –allows you save the Expression Tree and exit the XACML Expression window.
- Attribute buttons – allow you to add XACML attributes to the Expression Tree. Only the XACML attributes declared in the current Access Control Domain are available. There is a button for each category of XACML attribute: subject, resource, action and environment attributes.
- Named Expression button –allows you to add a named expression to the Expression Tree.
- Font Setting button – allows you to change the font for the attributes, values, functions and named expression displayed in the text pane.
- Function buttons– allow you to add a function to the Expression Tree. There are eight function categories: Boolean, Relational, String, Arithmetic, Bag, Set, Date and Time, and Conversion.

The interface provides descriptions of individual functions through pop-up 'tool tips'.

A condition comprises expressions

Each rule has a condition comprising a set of expressions which are declared in an expression tree.

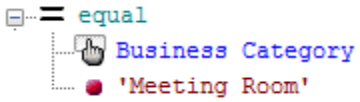
The condition for the first rule in this tutorial has the following expressions:

resource has attribute Business Category = 'Meeting Room' AND
subject has attribute User Name = 'asherma' AND
(Action = 'ModifyEntry' OR Action = 'AddType' OR Action = 'RemoveType' OR
Action = 'AddValue' OR Action = 'RemoveValue')

Every expression has a function and XACML attributes. The first expression is:

resource has attribute Business Category = 'Meeting Room'

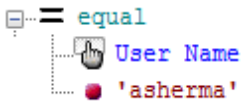
The function is equal and the XACML attribute is Business Category, and is represented in the expression tree as follows:



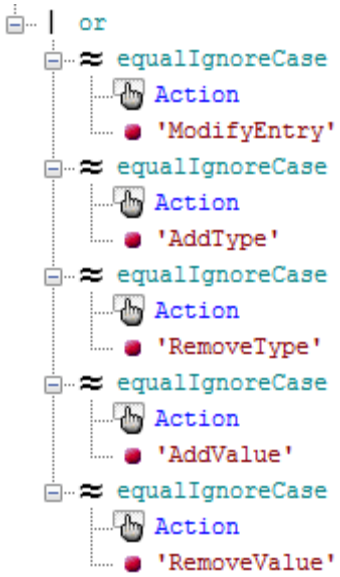
The second expression is:

subject has attribute User Name = 'asherma'

The function is equals and the XACML attribute is User Name, which is represented in the expression tree as follows:



The remaining expressions are represented in the expression tree as follows:

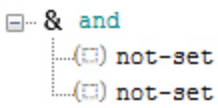


The three expressions are tied together with a single Boolean AND function.

Defining the condition

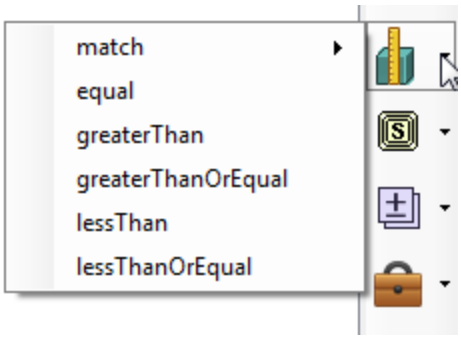
To define the first expression in the rule's condition:

1. The three expressions in the rule's condition are combined by a Boolean 'And' function: In the **XACML Expression** window, drag and drop the **&** from the **Functions Dashboard** to the node at the top of the expression tree. The function is displayed in the expression tree with two empty nodes below it.

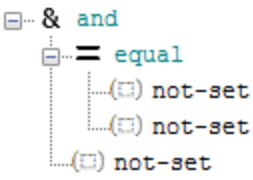
   

   To replace a function, drag and drop another function on top of the function to be replaced.

2. You can now start to define the first expression in the condition: Click the **Relational Functions** button. A list of functions is displayed.

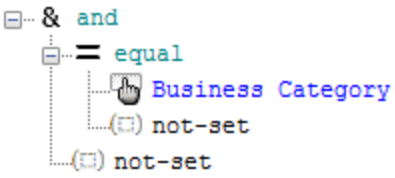


3. Drag and drop **= equal** onto the first not-set node in the expression tree. The equal function is added to the tree with two new empty nodes below it.



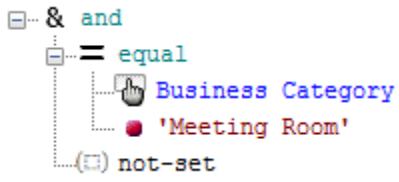
4. Click the **Resource Attributes** button.

5. Drag and drop **Business Category** onto the first not-node below the equal function.



6. Double-click the **not-set** node below Business Category. The String Editor window is displayed.

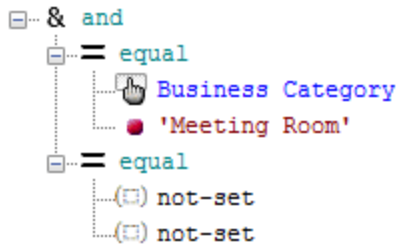
7. In the **Value** box, enter *Meeting Room* and then click **OK**. The string is added to the expression.



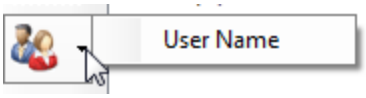
Now define the second expression.

Defining the second expression

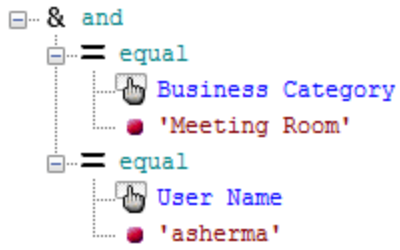
1. From the **Functions Dashboard**, drag and drop the **= equal** function onto the remaining **not-set** node. The equal function is added to the tree with two new empty nodes below it.



2. Click the **Subject Attributes** button.



3. Drag and drop **User Name** onto the first **not-set** node below the equal function.
4. Double-click the **not-set** node below User Name. The String Editor window is displayed.
5. In the **Value** box, enter *asherma* and then click **OK**. The string is added to the expression.

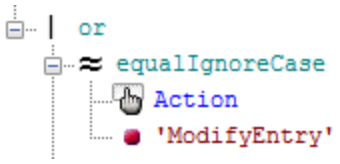


Now define the remaining expression.

Defining the remaining expression

1. Right-click the **& and** function at the top of the expression tree, then click **Add New Argument**. A new not-set node is added.
2. From the **Functions Dashboard**, drag and drop the **| or** function onto the new **not-set** node. The | or function is displayed with two new not-set nodes below it.
3. In the **Expression Tree**, right-click the **| or** function, and then click **Add New Argument**. A new not-set node is displayed below the function.
4. Repeat step 3 until there are five not-set nodes below the | or function.

5. From the **Functions Dashboard**, drag and drop the **= equal** function onto the first **not-set** node below the | or function. The equal function is added to the tree with two new empty nodes below it.
6. Click the **Action Attributes** button. The XACML attribute Action is displayed.
7. Drag and drop **Action** onto the first not-set node below the equal function.



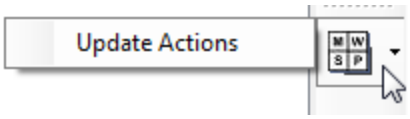
8. Double-click the **not-set** node below Action. The XACML Value (Enumerated) window is displayed.
9. Choose **ModifyEntry** from the **Value** dropdown list and click **OK**.
10. Repeat steps 5 through 9 to add the following to the expression tree:
    - Action = AddType
    - Action = RemoveType
    - Action = AddValue
    - Action = RemoveValue

Working with named expressions

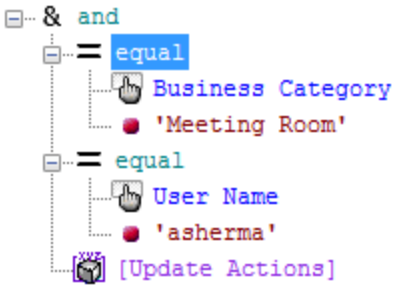
A named expression is an expression that is saved and can then be reused in different rules. If you modify a named expression, then the change will affect every rule it appears in.

These steps are not required to define the first rule, but are included in this tutorial to familiarise you with named expressions:

1. Right-click the **| or** function.
2. Click **Save as a Named Expression**. A window is displayed.
3. Enter *Update Actions* and then click **OK**.
4. Right-click the **| or** function, then click **Delete**. The node is deleted from the tree.
5. Right-click the **& and** function at the top of the expression tree, then click **Add New Argument**. A new not-set node is added.
6. Click the **Named Expressions** button. The named expression you just created is displayed.



7. Drag and drop **Update Actions** onto the not-set node in the expression tree.



You can view the text version of the named expression by clicking on it and hovering your mouse over it.

Alternatively, to edit the named expression, right-click it and select **Modify Named Expression** from the menu. Click **Yes** to confirm you want to proceed. The XACML Named Expression window is displayed. To view the named expression in a tree format click the **Edit** button.

8. Click the **Save and Exit** button. The XACML Expression window closes and the condition is displayed in the Condition box of the XACML Rule window.
9. Click the **Save** button. The rule is added to Policy Versions tab.
10. To view the named expression:
    a. In the right pane, click the **Policy Versions** tab.
    b. In the first filter box, click **Named Expressions**. The named expressions are listed in the summary area of the tab.
    c. Click the named expression and then click the **Open** button. The XACML Named Expression window is displayed.
    d. Click the **Edit** button. The named expression is displayed in the XACML Expression window.

Define the second rule

The second rule's condition is as follows:

Action = 'ReadEntry' OR Action = 'BrowseEntry' OR Action = 'ReturnDN' OR
Action = 'ReadType' OR Action = 'FilterMatchType' OR
Action = 'ReadValue' OR Action = 'FilterMatchValue' OR
Action = 'DiscloseEntryOnError' OR Action = 'DiscloseTypeOnError' OR
Action = 'DiscloseValueOnError'

To define the second rule:

1. Right-click in the **Policy Versions** tab and select **New Rule** from the menu.The XACML Rule window is displayed. It allows you to define a new rule for the currently selected policy.
2. In the **Label** box, enter *Search & Read access control*.
3. In the **Description** box, enter a short description of the rule, such as *Permit all users search and read access to all entries*.
4. Click the **Edit** button. The XACML Expression window is displayed.
5. Drag and drop the **| or** function from the **Functions Dashboard** to the **not-set** node at the top of the Expression Tree. The function is added to the expression tree with two empty nodes below it.
6. In the **Expression Tree**, right-click the **| or** function, then click **Add New Argument**. A not-set node is added to the tree.
7. Repeat the above step until there are ten not-set nodes.
8. From the **Functions Dashboard**, drag and drop the **= equal** function onto the first **not-set** node below the | or function. The equal function is added to the tree with two new empty nodes below it.
9. Click the **Action Attributes** button. The XACML attribute Action is displayed.
10. Drag and drop **Action** onto the first **not-set** node below the equal function.
11. Double-click the **not-set** node below Action. The XACML Value (Enumerated) window is displayed.
12. Choose **ReadEntry** from the **Value** dropdown list and click **OK**.
13. Repeat steps 9 through 12 in order to add the following to the remaining nine not-set nodes:
    - Action = BrowseEntry
    - Action = ReturnDN
    - Action = ReadType
    - Action = ReadValue
    - Action = FilterMatchValue
    - Action = FilterMatchType
    - Action = DiscloseEntryOnError
    - Action = DiscloseTypeOnError
    - Action = DiscloseValueOnError
14. Click the **Save and Exit** button, followed by the **Save** button.

Activate the policy

For a policy to take effect it must be activated. Only one version of a policy can be active at any time. This ensures that after writing a new version of a policy, you can activate it at an appropriate time and also have the option to roll back by activating the previous version if necessary.

To activate the policy:

1. Click the **Policy Versions** tab.
2. Click the **Version Management** button followed by **Activate**. A warning is displayed.
3. Click **Yes**. The policy's Status is now *active,open*.

This signifies that the rule is in use (active) but can still be modified (open).

Test the policy

You can test the policy by attempting to modify a meeting room entry through Access Presence, first as Andrew Sherman and then as another user. (For the instructions to configure for Access Presence, see Configuring for Access Presence).

To test the policy:

1. Open the following URL:

   http://[hostname]/ViewDS/Deltawing/webdua.cgi?

2. Log on with the user name **asherma** and password **testpass**.
3. In the drop-down box, click **Function Search** and then click **Access**. The Advanced Search page is displayed.
4. In the **Function** box, enter *meeting room* and press **Enter**. A list of meeting rooms is displayed.
5. Click the third meeting room in the list. The entry for the Sales Meeting Room is displayed.
6. Click **Modify**. The Modify page is displayed.
7. Modify the contents of the **Description** box and then click **Save**.
8. Log off by closing the browser session.
9. Repeat this task from step 1, logging on with the user name **rturnbu** and password **testpass**. This user will not be able to modify any entries.

To see a trace of the authorization decision request and response, see the ViewDS Management Agent help topic.

Lock the policy

Once you lock a policy you cannot delete or modify it. You can, however, create a new policy based on an existing policy by clicking the New button in the Policy Versions tab.

To lock the policy:

1. Click the **Policy Versions** tab.
2. Click the **Version Management** button followed by **Lock**. A warning is displayed.

3. Click **Yes**. The policy's Status is now *active,locked*.

Key concepts – Distribution and replication

This section introduces you to directory distribution and replication, and includes high-level guidance to help you adapt ViewDS to your requirements.

It covers the following topics:

- Understanding distribution and replication
- Distributing or replicating a DIT

Understanding distribution and replication

A DIT can be distributed or replicated to other DSAs.

A DIT might be distributed so that different organizations can manage their own data while still allowing access by another associated organization. This might be desirable when, for example, a company is spread across different countries with a different directory in each.

Some or all of the data in a DSA's DIT can be replicated to other DSAs to provide, for example, fail-over, load balancing or public access to part of a DIT.

The key concepts are:

- Peer trust
- Replicating
- Distributing

Peer trust

Before a DIT, or part of a DIT, can be replicated or distributed between DSAs, both DSAs need peer information about each other. The peer information tells each DSA which other DSAs it can trust, and the incoming and outgoing levels of trust.

For example, consider an implementation with two DSAs. The first DSA, DSA X, is within a firewall, and part of its DIT is replicated on DSA Y. This second DSA is outside the firewall and provides public access to the replicated area of the DIT.

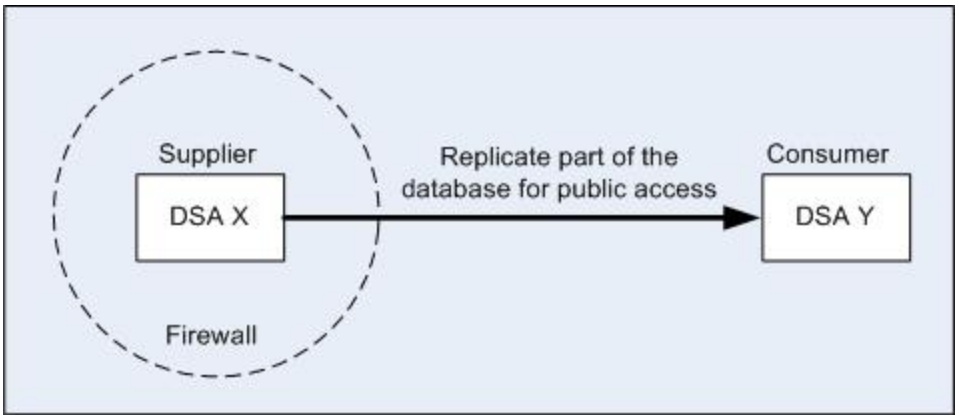


Figure 23: Supplier and Consumer DSAs

In this scenario, DSA X does not trust DSA Y to authenticate access to its data. DSA Y, however, does trust DSA X to authenticate access to its copy of the data.

Replicating

A replication agreement comprises a supplier DSA and a consumer DSA. The supplier DSA provides a shadow copy of its data to the consumer DSA. In Figure 23: Supplier and Consumer DSAs, DSA X sends a shadow copy of its data to DSA Y, the consumer.

DSA Y can service queries on its copy of the data, but cannot modify it directly. When a user modifies data through DSA Y:

1. DSA Y sends the modifications to DSA X, which updates to its database.
2. DSA X sends the modifications back to DSA Y to be replicated in its database.

ViewDS implements the X.500 Directory Information Shadowing Protocol (DISP).

Distributing

There are two types of distribution:

- Collaboration
- Delegation

Collaboration

Consider two DSAs, DSA X and DSA Y:

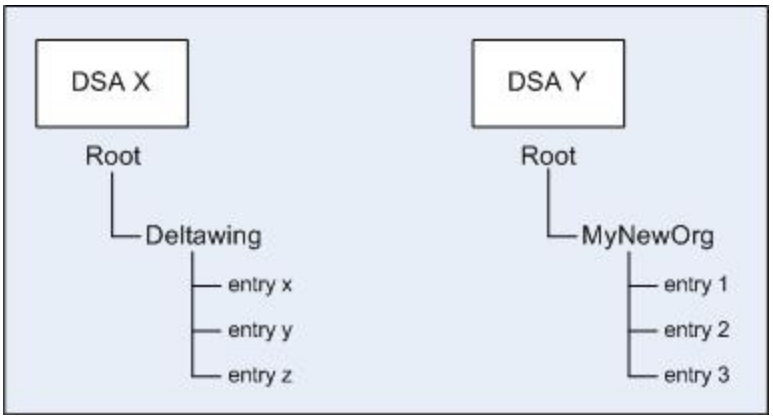


Figure 24: Example DSAs

With collaboration, users would see the following DIT:

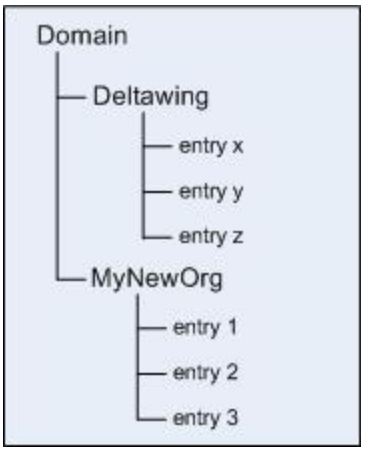


Figure 25: DIT distributed through collaboration

Setting this up involves adding an entry to each DIT that contains a knowledge reference. Each knowledge reference points to the other DSA (see Figure 26: DSAs in a collaboration below).

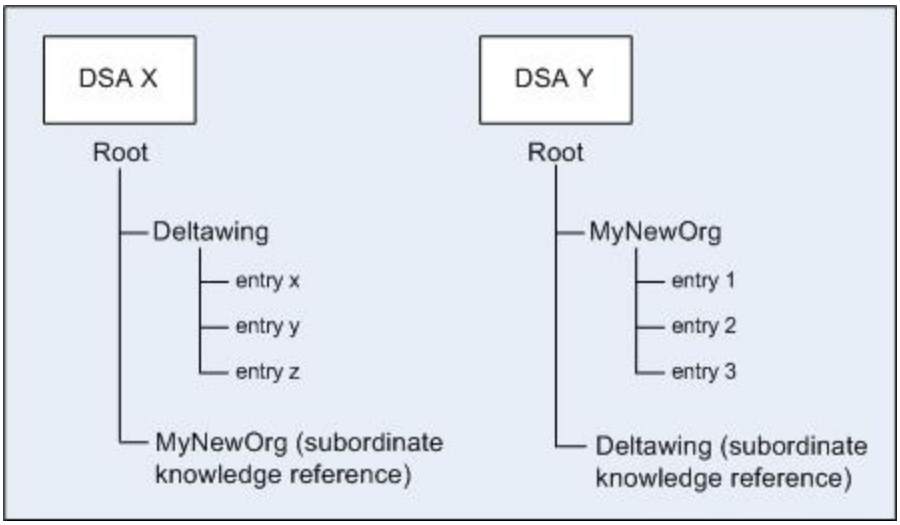


Figure 26: DSAs in a collaboration

Delegation

Using the same example DSAs, a delegation provides the following DIT to users:

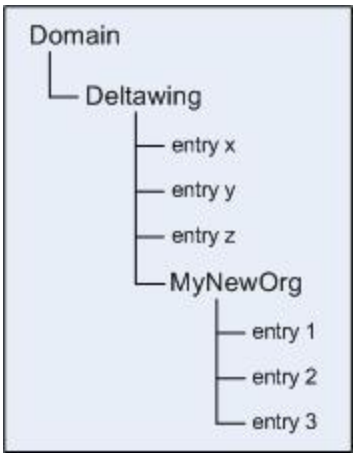


Figure 27: DIT distributed through delegation

Setting this up involves making DSA X the superior DSA by adding an entry to its DIT that contains a subordinate knowledge reference. When you create this entry, ViewDS Management Agent automatically configures the rest for you:

- ViewDS Management Agent makes DSA Y the subordinate DSA by adding an entry to its DIT that contains a superior knowledge reference.
- ViewDS Management Agent positions the knowledge reference in the DIT so that the hierarchy reflects that of the DIT of the superior DSA.

The configuration is shown below; each knowledge reference points to the other DSA's DIT (see Figure 28: DSAs in a delegation below).

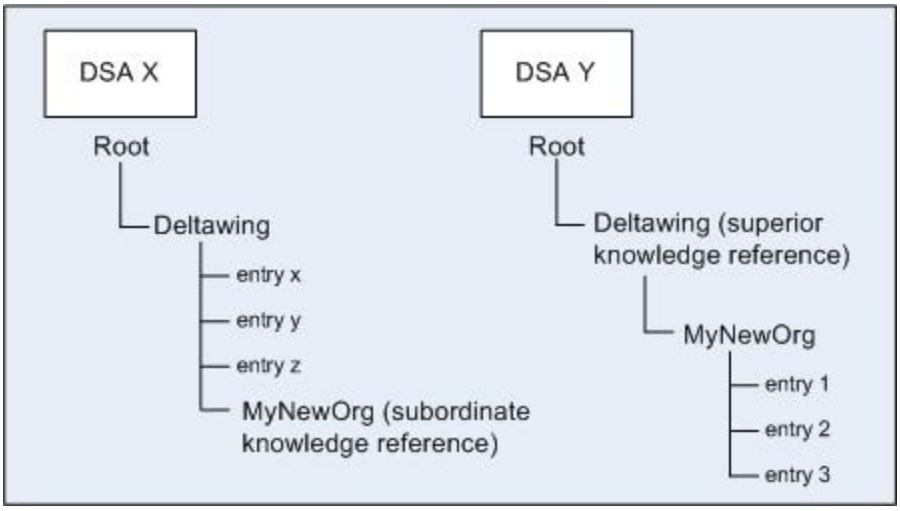


Figure 28: DSAs in a delegation

Distributing or replicating a DIT

For the steps to distribute a DIT, see the topic Distribute a DIT in the ViewDS Management Agent help. In brief:

1. Configure the DSAs for distribution.
2. Define peer trust between the DSAs.
3. Determine which type of distribution you will be using, collaboration or delegation.
4. Create knowledge references for each DSA.

For the steps to replicate a DIT, see the topic Replicate a DIT in the ViewDS Management Agent help. In brief:

1. Configure the DSAs for replication.
2. Define peer trust between the DSAs.
3. Create a replication agreement between the DSAs.