7

Configuring External Authentication

This chapter discusses adding external authentication. External authentication allows users—other then the default admin—to login and use IS1200 services.

External Authentication can be configured using either *Web-Admin* or the *Command Line Interface* (CLI). Once configured, administrators—and other users—signing in to IS1200 are authenticated by standard network external services such as *Active Directory* (AD) or *Network Information Services* (NIS). This keeps the IS1200 secure when multiple users access its services.

Topics include:

Overview of External Authentication

When the IS1200 is first installed, only two users—`root` and `admin`—can login and use the system. In normal use, other users may need access to the IS1200. Ideally a standard, IT administered, network authentication system is the best solution to validate these other users.

Setting up *external* authentication ensures only authorized users and applications can access the IS1200, and allows centralized management of the user database.

Besides controlling user login, external authentication can also control file access—when displaying search results or applying actionable services—through policy groups. To grant or deny file access privileges to a user, or group, the IS1200 needs to be able to identify and authenticate them. For more information, see "Policy Groups: Authorization Policies" on page 183.

The IS1200 supports external authentication using:

- *Network Information Services* (NIS)
- *Active Directory* (AD) using AdvancedAD (Kerberos) protocols

When external authentication is configured, the server does the following:

- Checks the appropriate authentication server to validate all users that login.
- If access checking is enabled (see "Controlling ACL Checking" on page 447 and "Turning Search ACL Checks ON or OFF" on page 432 for more information), the system filters search results based on the user login ID, the policies you set, and the file permissions. For information on policies, see "About Policies and Policy Groups" on page 184.

In addition, configuring external authentication enables you to view user and group names when you create audit reports and coalescence reports.

Note: Authentication configuration is not saved when server configuration is backed up. After restoring a system configuration, you must reconfigure all external authentication servers. See "Administration:

Backup, Licenses, and Health" on page 309 for more information.

The Authentication Services Listing Display

To manage Authentication Services, from the *Web-Admin* navigation pane under *Authentication*, click **Dir. Server Dashboard**.
The *Authentication Manager* page appears.

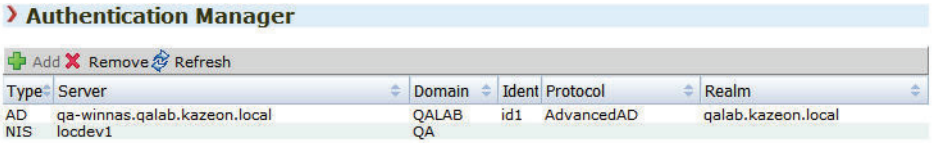


Figure 23 Authentication Manager Pane

The listing below the toolbar displays the currently configured authentication services and the following information for each:

Type. Displays authentication directory type (NIS or AD).

Server. The name of the NIS or AD server.

Domain. The NIS or AD domain name.

Identity. The identity name (from the Identity Vault) used to access this service.

Protocol. The type of authentication protocol used to communicate with the authentication service. For example, *AdvancedAd*, or *Kerberos*.

Realm. The Kerberos realm the authentication server applies to.

This field is only used for AdvancedAD and Kerberos protocols.

Managing Authentication for Network Information Services

Web-Admin can configure, view, and remove authentication for *Network Information Services* (NIS). Only one NIS server may be added.

Adding NIS external authentication using

1. From the *Web-Admin* navigation pane, click **Add NIS Server** under *Authentication* to go directly to the dialog in the next step, or click **Dir. Server Dashboard** to open the *Authentication Manager* pane.

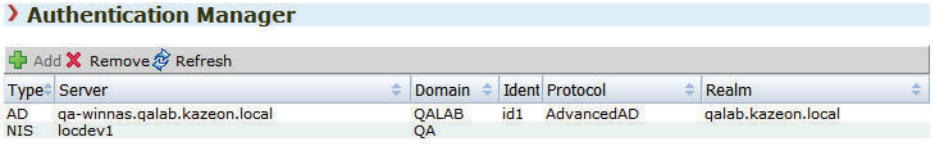
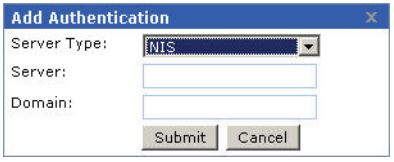


Figure 24 Authentication Manager Pane

2. From the *Authentication Manager* pane, click **Add** in the toolbar. The *Add Authentication dialog* opens.



3. In the *Add Authentication* dialog, select **NIS** from the *Server Type* drop-down list
4. Specify values for the following fields:

   Server. Enter the name of the NIS server to add.

   Domain. Enter the name of the NIS domain.

5. **Submit**. Click to add the authentication service.

Any existing NIS external authentication configuration is replaced when a new NIS authentication service is added.

To remove external authentication for NIS using *Web-Admin*

1. From the listing below the toolbar of the *Authentication Manager* page, select the authentication configuration to remove.
2. From the tool bar, click **Remove**.
   The system removes the authentication configuration.

To configure NIS external authentication using the CLI

1. Login to the CLI as admin.
2. To add NIS authentication with a username, enter the following command:

   ```
   add authentication nis domain <domainName> server <serverName>
   ```

   Where:

   `<domainName>` is the hostname or IPv4 address of the NIS domain
   `<serverName>` is the hostname or IPv4 address of the server hosting the NIS service

Managing Authentication for Active Directory

Both *Web-Admin* and the *Command Line Interface* can be used to configure, view, and remove external authentication for *Active Directory* (AD) services. The IS1200 supports AD servers on Windows 2000 and Windows 2003. Use the following guidelines to successfully configure authentication for AD:

- There must be both an operating DNS server and an AD server. Typically, they are the same server.
- For Windows 2000, authorized users must either be members of the Administrators group or have explicit rights to add clients to the domain.
- The following ports must be open on the AD server to insure all the communications necessary between the AD server and the IS1200.

Table 11 Active Directory Ports Required to be Open for IS1200 Communications

Service	Port	UDP	TCP	Comments
LDAP	389	X	X	UDP – Get Site Info
Kerberos	88	X	X	
KPassword	464	X	X	TCP – Used for Big PACs
SNTP	123	X		Optional Time Sync
DNS	53	X	X	TCP – Used for Big responses
SMB MS-RPC	445		X	
SMB	139		X	Older NTLM pass-through authentication
HTTP	80		X	
GC	3268	X	X	Global catalog lookups

For Windows 2003, authorized users must either have explicit rights to add clients to the domain, or they must be members of the *Account Operators* group, *Domain Power Users* group, or *Administrators* group.

Only one AD server may be added. Any existing AD external authentication configuration is replaced when a new AD authentication service is added.

Specifying AD Server Identities

When configuring Active Directory (AD) external authentication, an account (a username and password from an AD server) must be provided to the IS1200 to use when communicating with the AD server. That account must have sufficient privileges to allow the IS1200 to join the AD domain. Usually this means that account has administrator status.

Joining an AD domain enables the IS1200 to use the AD server to authenticate other users that log in to the IS1200 to search for information or create reports. It also allows the system to list user names in reports and to filter search results based on authorization rules or file permissions.

AD authentication is configured from the CLI or *Web-Admin* using identities.

An *identity* is an AD account (user name and password) stored in the IS1200 *Identity Vault* under an identity name. That identity supplies the IS1200 with the necessary username and password whenever the server accesses the AD server. Using identities is required for security reasons with AD authentication. For information on how to set up an identity, see "Adding Identities to the Identity Vault" on page 89

Note: The system stores all identities in the *Identity Vault* in an encrypted form using the cluster key. If the cluster is changed, all identities must be re-added.

Active Directory Server Protocols Supported

The IS1200 supports Windows domain AD authentication for Windows operating systems using the following protocols:

- *Advanced AD* - An implementation of AD using Kerberos that does not require fully-qualified host names. When configured (along with the correct DNS settings), it allows automatic server discovery when only the domain is provided while adding new authentication servers.
- *Kerberos* - The Kerberos version 5 authentication protocol is the default for network authentication on Windows 2003 computers.

  Kerberos uses fully-qualified domain names.

Windows 3.11, Windows 95, Windows 98, and Windows NT 4.0 use the NTLM protocol for network authentication in Windows 2000 domains. Computers running Windows 2000 use NTLM when authenticating with servers using Windows NT 4.0 and when accessing resources in Windows NT 4.0 domains. However, the protocol of choice in Windows 2000 is Kerberos version 5.

Additional Requirements for AD Kerberos Authentication

AdvancedAD and Kerberos authentication have the following additional system requirements:

For both AdvancedAD and Kerberos, one DNS server in your organization must support reverse DNS lookups:

- If a DNS server is already configured in your IS1200 `/etc/resolv.conf` DNS name resolution file, the IS1200 uses this server. DNS settings should be pre-configured using the `/sbin/kaz_setup.pl` initialization script.
- If the Windows AD server is also a DNS server (supporting reverse DNS lookups), the IS1200 adds an entry to `/etc/resolv.conf` for the Windows AD server.
- If neither a) nor b) above apply, then `/etc/resolv.conf` must have an entry added for a DNS server that supports reverse DNS lookups.

  The entry should be similar to the following. Substitute the IP address in the following example with an IP address provided by your system administrator:

  Example: `nameserver 10.11.12.13`

For Kerberos only, the IS1200 hostname must be a fully-qualified host name and its domain name must match the *Windows 2000 AD* server domain name. (From the CLI, type hostname `-f` to view the current

IS1200 host name and ensure it is fully qualified—for example `g11.kazeon.local` and not a short version like `g11`.) If the IS1200 hostname is a short name, run the `kaz_update.pl` script to fix the host name. See "Authentication Problems" on page 473 for more details on using `kaz_update.pl`.

Active Directory Authentication Procedures

The following procedures are available to manage AD authentication on the IS1200:

Adding AD Authentication Using *Web-Admin*

1. From the *Web-Admin* navigation pane under *Authentication*, click **Add AD Server** to go directly to the dialog box in the following step, or click **Dir. Server Dashboard** to open the *Authentication Manager* pane and then click **Add** in the tool-bar.

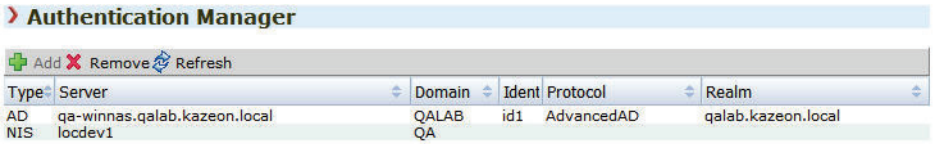


Figure 25 Authentication Manager Pane

2. **Server type**. Select *Active Directory* from the drop-down menu.

3. **Server**. Enter the name of the AD server. When *AdvancedAD* is configured, the server name is optional, if omitted AdvancedAD attempts to automatically discover the server name. The IP address of the DNS server must be correctly configured for this feature to function.

4. **Domain**. Enter the *Active Directory* domain name.

5. **Identity**. Select an identity from the drop-down menu for the IS1200 to use when accessing the AD server. For more information, see "The Identity Vault" on page 87.

Note: The username stored in the identity must have sufficient privileges to allow the IS1200 to join the AD domain. Usually this means that account has administrator status.

6. **Container**. Optional, container name for the *Active Directory* organizational unit, with hierarchical containers separated by slashes. See "Support for Multiple Organization Units (OUs)" on page 84 for more details.
7. **Submit**. Click to add the authentication service.

Note: To replace an existing AD Authentication, you must first remove the current AD Authentication and then add a replacement.

8. Proceed to "Verifying Active Directory Configuration" on page 76 and verify the AD configuration.

To configure AD external authentication using the CLI

1. Log in to the *Command Line Interface* as admin.
2. Enter the following command:

```
add authentication active-directory domain
  <domainName> [server <serverName>] identity
  <identityName>
```

Where:

`<domainName>` is the hostname of the AD domain
`<serverName>` is the hostname of the server hosting AD, [optional for AdvancedAD]
`<identityName>` is an identity already stored in the IS1200 *Identity Vault*.

3. Proceed to "Verifying Active Directory Configuration" (immediately below) and verify the AD configuration.

Verifying Active Directory Configuration

After configuring AD authentication from *Web-Admin* or the *Command Line Interface*, verify the configuration. In the examples below, the AD server name is `qa-winnas` with IP address `10.10.140.3`, and the domain name is `qalab.kazeon.local`. The hostname for the IS1200 is `myok1`, and the IP address is `10.10.140.100`. Identity is `myidentity`. The procedure used depends on the authentication protocol required. Refer to the appropriate procedure below.

Verifying Advanced AD and Kerberos Protocol

To verify AdvancedAD using Kerberos protocol, do the following:

1. (*For AD Kerberos only*)

   Ensure the hostname `-f` command resolves to a fully qualified domain name (FQDN) (if it does not, run `kaz_updatehosts.pl` to fix the host entry).

   For example:

   ```
   # hostname -f
   myok1
   # /opt/openkaz/bin/kaz_updatehosts.pl
   Setting fully-qualified hostname as myok1.kazeon.local
   # hostname -f
   myok1.kazeon.local
   ```

2. Ensure the AD server is ping-able, using both the short domain name, the FQDN, and that both ping to the same IP address. In the example below `qa-winnas`, `qa-winnas.qalab.kazeon.local`, and `10.10.140.3` are all ping-able:

```
[root@myok1 root]# ping qa-winnas
PING qa-winnas.kazeon.local (10.10.140.3) 56(84) bytes of data.
64 bytes from qa-winnas.qalab.kazeon.local (10.10.140.3): icmp_seq=1 ttl=127
   time=0.145 ms
64 bytes from qa-winnas.qalab.kazeon.local (10.10.140.3): icmp_seq=2 ttl=127
   time=0.139 ms
--- qa-winnas.kazeon.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.139/0.142/0.145/0.003 ms
[root@myok1 root]# ping qa-winnas.qalab.kazeon.local
```

```
PING qa-winnas.kazeon.local (10.10.140.3) 56(84) bytes of data.
64 bytes from qa-winnas.qalab.kazeon.local (10.10.140.3): icmp_seq=1 ttl=127
   time=0.145 ms
64 bytes from qa-winnas.qalab.kazeon.local (10.10.140.3): icmp_seq=2 ttl=127
   time=0.139 ms
--- qa-winnas.kazeon.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.139/0.142/0.145/0.003 ms
```

3. Ensure the DNS server supports forward DNS lookups for both the short host name and the fully-qualified host name and both resolve to the same IP address. In the example below, using the DNS lookup command host for the AD server `qa-winnas` and `qa-winnas.qalab.kazeon.local` both resolve to `10.10.140.3`. In this example the short domain name may actually resolve to a different domain (`qa-winnas.kazeon.local` instead of `qa-winnas.qalab.kazeon.local`), but this is acceptable as long as the forward DNS lookup using the fully-qualified host name of `qa-winnas.qalab.kazeon.local` also works:

```
[root@myok1 root]# host qa-winnas
qa-winnas.kazeon.local has address 10.10.140.3
[root@myok1 root]# host qa-winnas.qalab.kazeon.local
qa-winnas.qalab.kazeon.local has address 10.10.140.3
```

4. Ensure that the DNS server supports reverse DNS lookups for the IP address obtained from the DNS forward name lookup in the previous step. In our example, the forward DNS lookup in the step above returns IP address `10.10.140.3`, and so in this step, using the host command for `10.10.140.3` should resolve to the fully-qualified host name:

```
[root@ChyeLinok1 root]# host 10.10.140.3
3.140.10.10.in-addr.arpa domain name pointer qa-winnas.qalab.kazeon.local.
```

It is important to ensure that the reverse DNS name lookups resolve to exactly the above convention of `<server>.<domain>` where `<server>` is the name of the AD server and <domain> is the name of the AD domain. Therefore, in this example, if the reverse DNS lookup resolves to a different domain like `qa-winnas.kazeon.local` or `qa-winnas.kazeon.com`, Kerberos

will not work. The DNS administrator should add the correct entry to the database and before attempting AD configuration again.

5. Ensure the DNS Server is in the same AD domain as the AD server. Kerberos may not accept DNS query results from a DNS server in another un-trusted domain.

Troubleshooting Adding Authentication Errors

The following messages may appear when using the `add authentication` command:

Table 12 Add Authentication Trouble-shooting Errors

Message	Description
Active Directory domain join failed: Remote service not running	The remote Kerberos service is not running. Check that you have connected to a Windows Active Directory that supports Kerberos.
Active Directory domain join failed: User not found	The user name specified in the identity is not found in the Kerberos domain. The most likely cause for this error is that an invalid or non-existent user name is specified in the identity parameter of the add authentication command.
Active Directory domain join failed: Invalid domain name	Used for AdvancedAD authentication only. The domain name is invalid.
Active Directory domain join failed: Invalid domain specified	The domain is incorrect. Check to ensure that value entered in the domain parameter is correct. Use the host command to see if the specified domain can be resolved, and the ping command to see if it is accessible.
Active Directory domain join failed: Invalid domain or server specified	The Kerberos realm does not exist. Check to ensure that value entered in the domain parameter is correct. Use the host command to see if the specified domain can be resolved, and the ping command to see if it is accessible.
Active Directory domain join failed: Invalid server name	The server name is invalid.
Active Directory domain join failed: Cannot find domain	Used for AdvancedAD authentication only. The domain name cannot be found for the specified server. Check to see if the correct server and/or domain name is specified.
Active Directory domain join failed: incorrect username or password	Incorrect user name or password specified. Verify that the information given in the identity is correct. If the password for the identity is incorrect, replace the identity using the force parameter and enter the correct password before re-attempting the add authentication command.
Active Directory domain join failed: Invalid user name	Used for AdvancedAD authentication only. The user name cannot be found in the Kerberos database.
Active Directory domain join failed: Domain names do not match	Kerberos authentication failed. Type hostname -f to ensure that your hostname is a fully-qualified name, e.g. myhost.abc.com, instead of the short form such as myhost. If this is not the case, edit your /etc/hosts file to change your host name to a fully-qualified domain name. Also ensure that your domain name after the dot matches the DNS domain name as shown with the dnsdomainname command.
Active Directory domain join failed: AD Server LDAP signing not supported	Kerberos authentication failed. The AD Server is configured using high security settings not supported by Kerberos. Upgrade to AdvancedAD using the set authentication command and attempt the add authentication command again.

Table 12 Add Authentication Trouble-shooting Errors

Message	Description
Active Directory domain join failed: Host name in /etc/hosts must be fully-qualified for Kerberos join	Kerberos authentication failed because the host name in /etc/hosts is not fully-qualified. You can do one of the following:
Fix the entry in /etc/hosts using the /opt/openkaz/bin/kaz_updatehosts.pl script to make it fully-qualified, then attempt the add authentication command again.	
Upgrade to AdvancedAD using the set authentication command. This does not require a fully-qualified host name in /etc/hosts.	
Active Directory domain join failed: Domain names do not match	Kerberos authentication failed due to a disabled account error. Check to ensure that the domain name matches exactly the domain name in the AD server. For example, if the domain name in the AD server is corpjupiter.jupiter.com, you must specify your domain name as corpjupiter.jupiter.com, and not jupiter.com. The host command must also be able to correctly resolve the server name. For example, if the server name is adserver, the host command must correctly resolve adserver.corpjupiter.jupiter.com (and not another name like adserver.jupiter.com).
Active Directory domain join failed: Clock skew too great	Kerberos authentication failed because the clock skew is too great. Verify that the time on the IS1200 is within 5 minutes of the time on the AD Server. If this is not the case, set the time on the IS1200 to match that of the AD Server, or run NTP to synchronize the clock times.
Active Directory domain join failed: Remote service not running	The AD server is not running on the specified server. Check to ensure that the server name is correct, that it can be resolved using the host command to the correct IP address, and that the AD server is running on the remote node.
Server name server must be fully-qualified.	Unable to obtain the fully-qualified server name for AdvancedAD authentication. The most likely cause for this error is that Centrify authentication is attempted using a short domain name, and the DNS is not correctly configured to resolve the short domain name to the fully-qualified one. Ensure that the DNS settings are correct, and enter the fully-qualified domain name.
Domain name domain must be fully-qualified	Unable to obtain the fully-qualified domain name for AdvancedAD authentication. The most likely cause for this error is that Centrify authentication is attempted using a short domain name, and the DNS is not correctly configured to resolve the short domain name to the fully-qualified one. Ensure that the DNS settings are correct, and enter the fully-qualified domain name.
Active Directory domain join failed	The add authentication command has failed for a reason other than those listed above. To view the precise error message, use the command debug subsystem authentication level debug to turn on debugging for the policy management module, then execute the add authentication command again. Detailed logs are in /var/openkaz/log/policy_mgmt.log.
Identity <identity> not found	The identity specified in identity keyword of the add authentication command is not found. This error will most likely occur if you have mis-typed the identity name, or if you have not created the identity using the add identity command.

Table 12 Add Authentication Trouble-shooting Errors

Message	Description
You must first leave the <domain> domain before joining this domain.	The add authentication command is attempted when the IS1200 is already joined to another domain. The IS1200 can only be joined to one domain at a time. Use the remove authentication command to leave the current domain before joining the new one.
No identities found	There are no identities found. You have to create an identity using the add identity command prior to using that identity in the add authentication command.
Internal error getting identity information.	This signifies an internal error. Contact EMC support.

The following messages may be displayed after a `remove authentication active-directory` command:

Table 13 Add Authentication Trouble-shooting Errors

Message	Description
Active Directory authentication service disabled	Active directory authentication is successfully disabled as requested
Active Directory authentication service already disabled	Active directory authentication is already disabled. The most likely cause for this error is that the administrator typed the remove authentication active-directory command when active directory authentication has already been disabled.

The following messages may be displayed after a `test authentication active-directory` command:

Table 14 Add Authentication Trouble-shooting Errors

Message	Description
Active Directory domain membership valid: <domain>	Not currently a member of the <domain> domain or realm.

The following messages may be displayed after a `set authentication` command:

Table 15 Add Authentication Trouble-shooting Errors

Message	Description
You must first remove AD authentication before setting the protocol	The IS1200 is currently joined to an AD domain. The set authentication command can only be used when the IS1200 is not currently joined to a domain. Remove AD authentication using the remove authentication active-directory command before re-attempting the set authentication command.
An internal error has occurred: unable to set AD Authentication.	An internal error has occurred. Contact EMC Technical Support for assistance.

Policy Manager Debugging. Error messages like "`Active Directory domain join failed`" are used to present a more user-friendly version of the less commonly encountered error messages to the user. To see the precise cause of the error, you can turn on debugging to view the precise error message that is displayed by the policy management module. For example, the following command in the CLI can be used to turn on debugging for the policy manager:

```
debug subsystem authentication level debug
```

Enter the add authentication command again after turning on debugging. The debugging messages from the policy manager can then be found in

```
/var/openkaz/log/policy_mgmt.log.
```

To remove AD external authentication using *Web-Admin*

1. From the list beneath the toolbar of the *Authentication Manager* page, select the authentication configuration to remove.
2. From the tool-bar, click **Remove**.

   The system removes the authentication configuration.

Overriding the current AD communication protocol using Linux

The set authentication command allows administrators to force authentication communications to use a specific protocol, however, before using this command the administrator must be completely familiar with the configuration limitations detailed in "Managing Authentication for Active Directory"

Use the command as follows:

```
set authentication active-directory protocol <protocol>
```

Where:
`<protocol>` = advanced AD, kerberos, or none.

If the value is set to none, the server reverts to automatically determining the best authentication protocol.

Checking the current AD communication protocol using Linux

Linux administrators may check the current AD protocol using the following command.

```
sysprompt> show authentication protocol
```

System responds:

```
protocol
--------
advanced AD     (or kerberos, etc)
```

Managing AD authentication using the CLI

To change the current authentication server to a different server, the existing AD authentication must first be removed.

Do the following to change to a new AD server:

1. Log in to the server as `admin`.
2. To check the current authentication settings, enter:

   ```
   show authentication details
   ```

   The server responds:

   ```
   type  domain  identity  server     protocol  realm
   ----  ------  --------  ------     --------  -----
   AD    QALAB   id1       qa-winnas  Kerberos
   qalab.kazeon.local
   ```

3. To remove the current AD (or NIS) authentication, enter:

   ```
   remove authentication active-directory (or nis)
   ```

   The server responds:

   ```
   OK
   [220]  Active Directory authentication service disabled
   ```

4. To check that an AD (or NIS) service was removed, enter:

   ```
   show authentication details
   ```

   The server responds:

   ```
   OK
   [220]  No authentication servers configured.
   ```

5. To add the new AD (or NIS) authentication, see one of the following:

- "To configure NIS external authentication using the CLI"
- "To configure AD external authentication using the CLI"

Configuration Issues with Multiple Domain Controllers

For installations using AdvancedAD in domains with multiple *Domain Controllers* (DCs), procedures are available allowing administrators to limit the DCs the IS1200 uses for authentication. Errors may occur when a limited set of DCs have had "ports opened" to allow IS1200 access, and the IS1200 has no way of knowing which DCs to work with. Additionally, they occur on slow networks and when DCs are off-line for maintenance or other issues.

Errors reported in these situations appear in log entries similar the following.

```
Jan  8 18:12:57 foo1 adclient[13601]: DEBUG <fd:17 get object> base.bind.ad
   Connecting to dc3.testdomain.acme.com:389
Jan  8 18:12:57 foo1 adclient[13601]: DEBUG <fd:17 get object> base.bind.ldap
   dc3.testdomain.acme.com:389 fetch dn="" filter="(objectclass=*)"
Jan  8 18:13:02 foo1 adclient[13601]: DEBUG <fd:17 get object> base.osutil fetch
   : Can't contact LDAP server (reference base/ldapbind.cpp:151 rc: -1)
Jan  8 18:13:02 foo1 adclient[13601]: DEBUG <fd:17 get object> base.bind.ad
   Failed to connect to dc3.testdomain.acme.com:389: fetch  : Can't contact LDAP
   server
```

Use the following steps to restrict the IS1200 to specific DCs in a given domain:

1. Suspend any current basic or deep classifications.
2. If authentication was previously added, remove it.
3. Login as root user to IS1200.
4. Change directory (cd) to `/etc/centrifydc`
5. Edit the centrifydc.conf file as follows:
   You will find lines similar to the follow:

```
#
# Specify dc and gc hostnames if your DNS isn't configured correctly
# for AD.  This is not recommended for production systems, since AD
# automatically updates DNS with failover and replica systems and optimizes
# for your site location.  This is provided mostly for evaluation systems
# which are using Unix DNS and can't add the _ldap and _gc service records
#
# dns.dc.<domain.name>: <hostname> [hostname] ...
# dns.gc.<domain.name>: <hostname> [hostname] ...
```

```
#
# Example:
# dns.dc.acme.com: anvil.acme.com coyote.acme.com
# dns.gc.acme.com: roadrunner.acme.com
#
# Note the hostname must resolve in DNS or be entered in /etc/hosts
#
```

To restrict DC's `host1.acme.com` and `host2.acme.com` on a domain called `acme.com`, add the following line to those above and save the file on all nodes:

```
dns.dc.acme.com: host1.acme.com host2.acme.com
```

Now add authentication using AdvancedAD protocol.
(continued next page)

6. If user login (using AdvancedAD) takes a very long time and the user belongs to large number of groups, do the following.

   a. Login as `root` user.

   b. Change directory (cd) to `/opt/openkaz/etc/`

   c. Edit the file `nsswitch.conf.centrifydc` as follows:

   Replace the line:

   ```
   group:      files centrifydc compat
   ```

   With:

   ```
   group:      files compat
   ```

   If AD authentication has already been added using AdvancedAD, then modify one more file.

   a. Change directory (cd) to /etc

   b. Edit the file nsswitch.conf as follows:

   Replace the line:

   ```
   group:      files centrifydc compat
   ```

   With:

   ```
   group:      files compat
   ```

Support for Multiple Organization Units (OUs)

By default, an *Active Directory* (AD) server responds to a client computer login request, by joining the client to the *Organizational Unit* (OU) under the requested domain.

A more complex organization may need to have a more sophisticated structure and organize the computers into different domains, possibly in a hierarchy. For example, a holding company (*ACME Conglomerates*) may have business units each with their own domains (*BusUnit1, BusUnit2*, etc). Each business unit may also have it's own AD server and business hierarchy (with corresponding organizational units such as *Marketing*, *HR*, *Accounting*, etc). When a marketing client computer requests authentication from the BusUnit1 AD server, it may need to be placed in the *BusUnit1-Marketing* domain, rather then just a *BusUnit1* domain.

To do this, specify the `container` command to override the default OU. For example, to join a client to the *BusUnit1/Marketing* domain, use the `add authentication` command as follows:

```
add authentication active-directory
```

`domain`	The NIS or AD domain name. Specify the fully-qualified domain name for Kerberos (if applicable)
`server`	IPv4 or hostname of system running the authentication service (optional for AdvancedAD only)
`user`	User to connect to the domain as (active-directory requires either a user or an identity)
`identity`	A case insensitive unique identifier for an identity
`container`	The optional container name for the AD organizational unit, with hierarchical containers separated by slashes

For example, to join a client to the BusUnit1/Marketing domain using an identity:

```
add authentication active-directory domain adtest.com
   server ad140 identity adtestid container
   BusUnit1/Marketing
```

Organizational Unit Persistence

Note the following Windows AD server behavior with respect to OU's:

- After a client joins a certain domain (and OU), it will always rejoin the same domain regardless of the OU specified, unless the AD Server OU registration entry is specifically deleted. For example, if a client previously joined the *BusUnit1* domain, it will always rejoin the *BusUnit1* domain on subsequent log ins. Likewise, if a client joined the *BusUnit1/Marketing* domain, it will

rejoin the *BusUnit1/Marketing* domain on subsequent log ins even if the container option is not specified in subsequent `add authentication` commands.

- To force a client to join a different domain (different from its last log in), an administrator must log in to the Windows AD Server and delete the registration entry for the computer. For example, to join *JohnSmith* to the *BusUnit1/Accounting* domain after he has already joined *BusUnit1/Marketing*, delete the *BusUnit1/Marketing* entry by selecting that entry and clicking **Delete** as shown below:

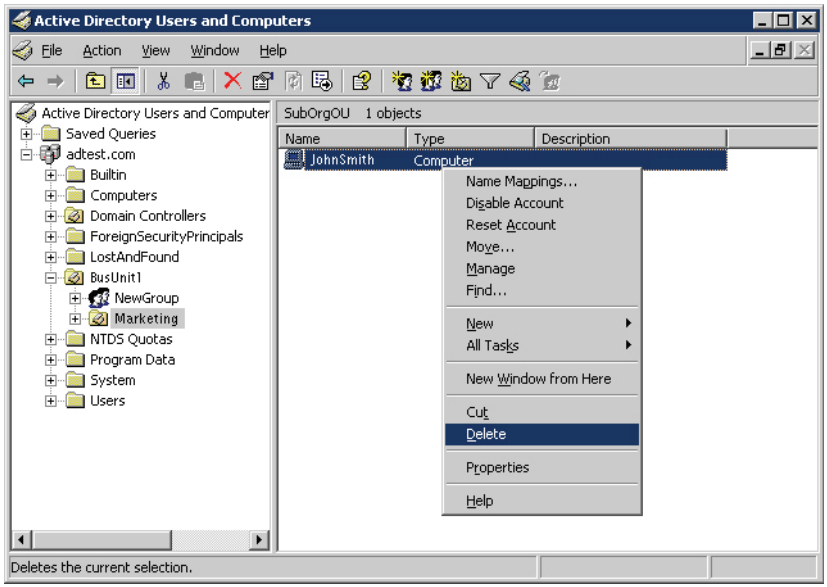


Figure 26 Active Directory Users and Computer Interface