



# Server Suite

Network Information Service Administrator's Guide

Version: 2024.x

Publication Date: 7/26/2024

## Server Suite Network Information Service Administrator's Guide

Version: 2024.x, Publication Date: 7/26/2024

© Delinea, 2024

### Warranty Disclaimer

DELINEA AND ITS AFFILIATES, AND/OR ITS AND THEIR RESPECTIVE SUPPLIERS, MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE INFORMATION CONTAINED IN THE DOCUMENTS AND RELATED GRAPHICS, THE SOFTWARE AND SERVICES, AND OTHER MATERIAL PUBLISHED ON OR ACCESSIBLE THROUGH THIS SITE FOR ANY PURPOSE. ALL SUCH MATERIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. DELINEA AND ITS AFFILIATES, AND/OR ITS AND THEIR RESPECTIVE SUPPLIERS, HEREBY DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO SUCH MATERIAL, INCLUDING ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT.

THE MATERIAL PUBLISHED ON THIS SITE COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN. DELINEA AND ITS AFFILIATES, AND/OR ITS AND THEIR RESPECTIVE SUPPLIERS, MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE MATERIAL DESCRIBED HEREIN AT ANY TIME.

### Disclaimer of Liability

IN NO EVENT SHALL DELINEA AND ITS AFFILIATES, AND/OR ITS AND THEIR RESPECTIVE SUPPLIERS, BE LIABLE FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES (INCLUDING LOSS OF USE, DATA, PROFITS OR OTHER ECONOMIC ADVANTAGE) OR ANY DAMAGES WHATSOEVER, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE, OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF SOFTWARE, DOCUMENTS, PROVISION OF OR FAILURE TO PROVIDE SERVICES, OR MATERIAL AVAILABLE FROM THIS SITE.

## Table of Contents

Network Information Service Administrator's Guide .....	i
<b>Network Information Services Administration .....</b>	<b>1</b>
Introduction to the Basics of NIS .....	1
Limitations of using NIS .....	1
Deciding to Maintain NIS in your Environment .....	2
Using the Network Information Service .....	2
How NIS Client Requests are Processed .....	2
Derived and Explicitly-defined Maps .....	3
Accessing NIS Maps in the Local Cache .....	3
Migrating Information from Existing Maps .....	4
Managing Automounts without Using NIS .....	4
Mounting Home Directories with the nosuid Option .....	6
Using Executable Maps .....	7
Testing the Status of the Automount Service .....	7
Testing the adauto.pl script results .....	7
Restarting the Automount Service .....	7
Distributing Automount Maps .....	8
Discontinuing Use of Legacy NIS Servers .....	9
<b>Preparing for Agentless Authentication for NIS Clients .....</b>	<b>9</b>
Deciding to Use Agentless Authentication .....	9
Planning for Agentless Authentication .....	10
Selecting a Zone to Use for NIS Authentication .....	12
Selecting a Computer for NIS Authentication .....	13
Configuring a Password Synchronization Service .....	13
Using Delinea Password Synchronization .....	13
Using Microsoft Password Synchronization Service .....	14
Locating Zones for Password Synchronization .....	15
<b>Configuring the Server Suite NIS server .....</b>	<b>15</b>
Installing the Centrify NIS Server .....	15
Adding IP Addresses from Which to Accept Requests .....	16
Starting the adnisd Process .....	17
Customizing the Update Interval for NIS Maps .....	18
Customizing the NIS Maps to PUBLISH .....	18
Configuring the Maximum Number of Map Sets .....	19
Handling Large Active Directory Groups .....	19
Splitting a single large group into multiple new groups .....	19
Setting the maximum length of new group names .....	19
Making the NIS Server Available .....	20
<b>Configuring NIS Clients .....</b>	<b>20</b>
Specifying the Server for NIS Clients to Use .....	20

## Table of Contents

Configuring NIS Clients on Linux .....	21
Configuring NIS Clients on Solaris .....	21
Configuring NIS Clients on HP-UX .....	22
Configuring NIS Clients on AIX .....	23
Verifying the Client Configuration .....	24
Checking the Derived passwd and Group Maps .....	24
<b>Importing and Managing NIS Maps .....</b>	<b>25</b>
Importing and Creating User and Group Profiles .....	25
Publishing Network or Custom Information .....	25
Importing Network NIS Maps .....	26
Creating New NIS Maps in Active Directory .....	27
Creating Maps for Common Network Services .....	28
aliases .....	28
audit_user .....	29
auth_attr .....	29
bootparams .....	30
ethers .....	30
exec_attr .....	31
hosts .....	32
netgroup .....	33
netmasks .....	33
networks .....	34
printers .....	35
prof_attr .....	35
project .....	36
protocols .....	36
rpc .....	37
services .....	38
user_attr .....	39
Creating Generic Custom Maps .....	40
Changing the Map Type .....	40
Maintaining Map Records in Active Directory .....	41
Modifying Map Records in Active Directory .....	41
Deleting a map stored in Active Directory .....	41
<b>Troubleshooting and Logging NIS Operations .....</b>	<b>42</b>
Analyzing Zones for Potential Issues .....	42
Verifying NIS Configuration for Servers and Clients .....	42
Updating the Startup Sequence .....	44
Using NIS Command Line Utilities .....	44
Configuring Logging for adnisd .....	44

# Network Information Services Administration

This section provides a brief overview of Network Information Services (NIS), including the basic advantages and limitations of using NIS to publish information. It also describes the Centrify solution for using NIS to respond to client authentication and lookup requests.

You should use this section to help you determine whether the Centrify Network Information Service (adnisd) is an appropriate solution for your organization's needs.

## Introduction to the Basics of NIS

In some environments, a Network Information Server (NIS) provides centralized storage and distribution of information that needs to be known throughout the network. In a typical NIS environment, one or more NIS servers are used to centrally manage a set of database **maps** that correspond to the system configuration files that are commonly found on UNIX systems. For example, there are NIS maps that correspond to the `/etc/passwd`, `/etc/group`, `/etc/hosts`, and `/etc/services` files. The maps provide the centralized information to a given set of computers that make up a **NIS domain**.

Each **NIS map** corresponds to a specific configuration file, such as the `/etc/passwd` or `/etc/hosts` file, and consists of a set of keys and values, and a version number for the data. When computers on the network require information stored in NIS maps, they send a **NIS client request** to the NIS listening port to query the **NIS server** for the information.

When a computer needs the information stored in a NIS map, it runs the `ybind` process to identify and connect to the NIS server best suited to respond to its requests. When the NIS server receives a request, it replies with the appropriate information from its set of NIS maps.

## Limitations of using NIS

Although NIS can be very efficient in responding to queries for network information, it is not a secure mechanism for providing authentication and authorization services. For example:

- If NIS clients use the broadcast service to locate NIS servers on the network, intruders can easily introduce their own NIS server with their own privileged accounts. Once a client binds to the rogue NIS server, the intruder can gain access to that client and perform unauthorized operations.
- The NIS server's only security policy is the `securenets` setting. The `securenets` setting identifies which NIS clients to accept queries from. If an intruder impersonates a client that the `securenets` setting allows the NIS server to accept, he can download all of the NIS data. Even if an intruder fails the `securenets` test, he could potentially inspect all of the NIS requests and decode the data to gain access.
- If NIS is used for authentication, password hashes are sent around the network in clear text and can be easily captured and cracked, making client systems vulnerable.

Because of these security risks, in most cases, you should plan to replace any legacy NIS environment with Active Directory as the central repository of identity information and the Centrify Agent for \*NIX (`adclient`) as the "client" requesting information. In some cases, however, it may not be practical or desirable to completely replace an existing NIS infrastructure. To handle those cases, Centrify provides its own Network Information Service (`adnisd`) that enables existing NIS clients to remain in place and co-exist with Active Directory.

## Deciding to Maintain NIS in your Environment

Active Directory and the Centrify Agent for \*NIX (`adcli`) provide more secure authentication, authorization, and directory services than provided by traditional NIS client-server communication. Therefore, when you install the Centrify Agent and join a domain, the Name Service Switch configuration file, `nsswitch.conf`, is normally modified so that account lookup requests are passed to Active Directory through the `adcli` process. This change to the `nsswitch.conf` file effectively bypasses the NIS client and server environment.

There are some situations, however, in which maintaining an ongoing or temporary NIS environment may be desirable or necessary. For example:

- If you have a legacy Network Information Server (NIS), you may have configured network information, such as `netgroup` or `automount` maps, that you want to make available in response to client requests.
- You may have applications that require access to a NIS server because they send requests directly to the NIS port and expect a NIS process to be listening there.
- You may have computers or devices, such as Network Attached Storage devices or computers with older or unsupported operating systems where you cannot install the Centrify Agent, that need access to information normally stored in NIS maps. Those computers or devices cannot join an Active Directory domain, but are capable of submitting NIS client requests. For those computers or devices, a NIS server may be the only option for providing authentication and look-up services.

If any of these scenarios apply to your organization, you may want to plan a deployment that includes the Centrify Network Information Service to complement the agent.

## Using the Network Information Service

To support computers and applications that are capable of submitting NIS client requests to a NIS server, the Server Suite provides its own Network Information Service. The Centrify Network Information Service, `adnisd`, is an optional process that can be installed on any computer where `adcli` is installed.

Once installed and running, the Centrify Network Information Service functions like a standard NIS server, but it responds to NIS client requests using the information stored in Active Directory, including any information imported from `passwd` and `group` NIS maps or from `/etc/passwd` and `/etc/group` files. The Centrify Network Information Service has some of the same security limitations as a standard NIS server, but it does allow you to provide encrypted authentication and directory service to computers where `adcli` cannot be installed.

The Centrify Network Information Service can be useful in environments where you plan a phased migration from existing NIS servers and clients or when the environment includes legacy systems that you cannot migrate or upgrade to support the Centrify Agent for \*NIX.

## How NIS Client Requests are Processed

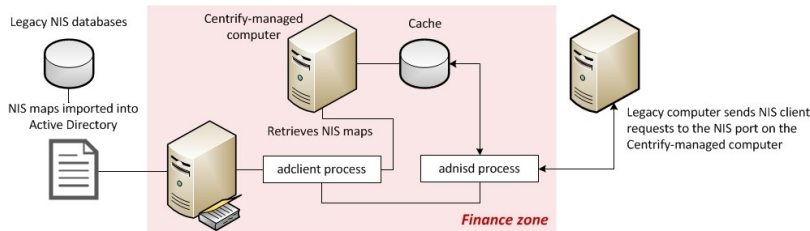
If you have decided to maintain a NIS environment, on either an ongoing or temporary basis, you can use the Centrify Network Information Service to replace existing NIS servers and the Access Manager console to migrate NIS map data to Active Directory.

The Centrify Network Information Service (`adnisd`) can run on any computer that has the `adcli` agent service installed. Computers that need access to the information stored in Active Directory can then be configured as NIS clients that send their NIS queries to the computer where both the `adcli` and `adnisd` service run.

## Network Information Services Administration

When `adnisd` receives a request from the NIS client, it checks its local cache of map data, then responds to the client that made the request. The local cache of map data is generated from the map data `adnisd` receives from Active Directory.

The following figure provides a simplified view of operation.



## Derived and Explicitly-defined Maps

Within the local cache, there are two types of maps: **explicitly-defined maps** and **derived maps**. Explicitly-defined maps are NIS maps imported into Active Directory from an existing NIS domain, imported from text files, or created manually using the Centrify Access Manager console. Derived maps are maps that are automatically generated from the information stored in Active Directory. Derived maps access the same data as the explicitly-defined maps using different keys. For example, the user and group maps in the local cache are not retrieved directly from Active Directory, but are generated based on the users and groups that have been enabled for the local computer's zone.

The maps derived from the zone information are `passwd.byname`, `passwd.byuid`, `group.byname`, and `group.bygid`. These automatically generated maps are placed in the local cache, and can then be used to look up or authenticate users by user name or by UID value, and groups by group name or by GID value. The Centrify Network Information Service also generates derived maps for explicitly-defined network maps that are stored in Active Directory. If `adnisd` finds a NIS map defined in Active Directory with a name it recognizes, such as `netgroup` or `services`, it automatically derives related maps. For example, a `netgroup` map will automatically generate the `netgroup.byhost` and `netgroup.byuser` maps. A `services` base map will generate the `services.byname` and `services.byservicename` maps.

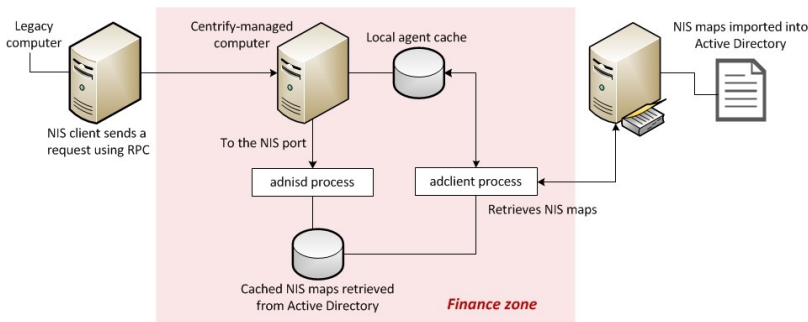
## Accessing NIS Maps in the Local Cache

Periodically, the `theadnisdprocess` connects to Active Directory through the `adclient` process to locate updates to explicitly-defined NIS maps. It then synchronizes the local cache of NIS map data to mirror any changes detected in Active Directory. After polling Active Directory for updates to explicitly-defined maps, the `adnisd` process retrieves all users and groups in the current zone from `adclient`, and generates the derived maps for user and group information.

In essence, the computer where both `adclient` and `adnisd` run acts as the NIS server for the local computer's zone. The NIS clients on the network communicate with `adnisd` using Remote Procedure Calls (RPC) sent to the NIS port on the Centrify-managed computer. The `adclient` process is responsible for all communication with Active Directory and maintains its own separate cache of data from which `adnisd` can derive the user and group information for the zone. The `theadnisdprocess` then stores all of the explicitly-defined and derived maps in its own local cache of map data (in most cases, `/var/centrifydc/nis/*`). Because `adnisd` always responds to NIS client requests using the data in its local cache, it can respond even when Active Directory is not available.

The following figure provides a simplified summary of operation.

## Network Information Services Administration



The `adnisd` process cannot be used with any legacy NIS servers in a NIS domain. It can only be used in conjunction with Active Directory and the Centrify Agent for \*NIX.

## Migrating Information from Existing Maps

If you have a legacy NIS environment, you can import user, group, and network information from existing NIS servers and domains. To import the information directly from an existing NIS server, you need to be able to access the NIS server and NIS domain from the Windows computer where the Access Manager console is installed. For example, if you have configured an existing NIS server to be accessible over the Windows network using Samba or a similar program, you can connect directly to that server and NIS domain to import maps. If the NIS server and NIS domain are not accessible from the Windows computer where the Access Manager console is installed, you should export the NIS maps to text files, then import the text files.



Importing existing maps simply provides a mechanism for migrating existing information to the Active Directory. Once the information is imported into Active Directory, the original maps are no longer used and the Centrify Network Information Service uses Active Directory to generate the maps it needs to service authentication requests.

For more information about importing existing user, group, or network information, see [Importing and managing NIS maps](#).

## Managing Automounts without Using NIS

If your primary reason for wanting to use NIS is to manage automount information, you have the option of storing the information in Active Directory then retrieving it through the `adnisd` process or directly through an LDAP request that bypasses the `adnisd` process.



The automount information must be stored in Active Directory. You can then choose whether to retrieve it using the Centrify Network Information Service (`adnisd`) or an LDAP query.

As an alternative to using the `adnisd` process, you can use the optional `adauto.pl` script located in the `/usr/share/centrifydc/etc` directory to get automount data. The `adauto.pl` script gets mount point information directly from Active Directory using LDAP. With the `adauto.pl` script, you can automount home directories using the information from NIS maps without running the `adnisd` server process.



## Network Information Services Administration

The `adauto.pl` script uses the information you store in the `auto.home` NIS map for the joined zone and any parent zones up the zone hierarchy from which the local computer inherits NIS map entries. Once you add the script to your automount configuration, the `automounter` program invokes the script and passes it the user name of the user logging on. The `adauto.pl` script then uses the `ldapsearch` command to retrieve the mount point information from Active Directory and returns the path to the remote home directory for the user logging on. The `automounter` will then attempt to connect to that home directory.

To use the `adauto.pl` script:

1. Add the appropriate `auto.home` mount points to Active Directory by importing or creating automount NIS maps.

For more information about importing existing `auto.home` or `auto_home` NIS maps, see *Importing Network NIS Maps*. For information about creating NIS network maps directly in Active Directory, see *Creating new NIS Maps in Active Directory*.

For example:

- Open Access Manager to navigate to a specific zone.
- Expand the zone to display NIS Maps.
- Select NIS Maps, right-click, then click **New > Automount**.
- Type `auto.home` or `auto_home` as the map name, then click **OK**.
- Select the new map, right click, then click **New** to add a new individual map record. For example, create a map record similar to this for all users in a zone:

Name: \* Network Path: `lmrh2:/home/&`

Comments: This is the automount path for users in this zone

2. If you are managing mount points on Linux or Solaris, edit the `/etc/nsswitch.conf` file to change the automount entry from `nis` to `files`. For example:

```
vi /etc/nsswitch.conf
...
automount: files
```

For other platforms, such as AIX, you can skip this step.

3. Verify the `adauto.pl` file is available in the `/usr/share/centrifydc/etc/` directory and is executable. For example:

```
ls -l /usr/share/centrifydc/etc/adauto.pl
total 1208
-rwxr-xr-x 1 root root 1921 Sep 27 10:37 adauto.pl
```

4. Create a symbolic link for `/etc/auto.home` or `/etc/auto_home` to the `adauto.pl` file. For example, on Linux computers:

```
ln -s /usr/share/centrifydc/etc/adauto.pl /etc/auto.home
```

On AIX computers, create the link to `/etc/auto_home`:

```
ln -s /usr/share/centrifydc/etc/adauto.pl /etc/auto_home
```

5. Edit the `/etc/auto.master` or `/etc/auto_master` file to call the `/etc/auto.home` file.

For example, on Linux computers add the following line to the `auto.master` file:

```
/export/home program: /etc/auto.home
```

## Network Information Services Administration

The specific syntax for the entry is different on different platforms. For example, not all platforms allow you to specify the program keyword in the `/etc/auto.master` file. For more information about the format of the entry, see the main page for `auto.master`. For example, on SuSE Linux, the entry should look like this:

```
/export/home /etc/auto.home
```

On SuSE Linux 10, the corresponding entry is:

```
/export/home program /etc/auto.home
```

On AIX and Solaris computers, add an entry like this to the `/etc/auto_master` file:

```
/export/home /etc/auto_home
```

On some platforms, you can invoke `automount` from the command line without editing the `/etc/auto.master` file. For example, you can invoke `automount` without editing the `/etc/auto.master` file by running a command similar to the following on Linux:

```
automount /export/home/ program /etc/auto.home
```

Command line mount points are not supported by `automount` on AIX.

6. Restart the `autofs` process. For example, on Linux:

```
service autofs restart
```

On AIX:

```
automount
```

On Solaris 10, the `automount` service is managed by the service management facility, `smf`, under the service identifier:

```
svc:/system/filesystem/autofs:default
```

You can use `svcadm` to perform administrative actions, such as stopping and restarting the service.

## Mounting Home Directories with the `nosuid` Option

To increase security when automatically mounting file systems, you might want to configure the `auto_home` or `auto.home` NIS map to prevent users from switching their user or group identity. You can prevent users from mounting file systems with a different user context by specifying the `nosuid` option.

To set the `nosuid` option in the `auto_home` or `auto.home` NIS map:

1. Open Access Manager to import or create a NIS map to be stored in Active Directory.
2. Expand the appropriate zone and the UNIX Data node to display NIS Maps.
3. Select NIS Maps, right-click, then click **New > Automount**.
4. Type `auto.home` or `auto_home` as the map name, then click **OK**.
5. Right-click the new map.
6. Click **New > Map entry** to add a new individual map record.
7. Set the fields in the map record similar to the following, to enable mounting of home directories with the `nosuid` option for all users in a zone:

```
Name: * Network Path: homeservername:/home/&
```

```
Options: -nosuid
```

You can use a similar approach to specify other or additional mount options—such as `noexec` and `nodev`—to the map entry.

### Using Executable Maps

On some platforms, local maps that have the execute bit set in their file permissions can be executed by the automount program and provided with a key to be looked up as an argument. The executable map is expected to return the content of an automount map entry on its `stdout` or no output if the entry cannot be determined. Direct maps cannot be made executable.

For more information about executable maps, see the main page for `automount`.

### Testing the Status of the Automount Service

After restarting the automount service, you can check the status of the service. For example, on Linux run the following command:

```
service autofs status
```

On all platforms, you can run the following command and check the output to verify automount operation:

```
/usr/sbin/automount -v
```

You should see output similar to the following:

```
automount: /export/home mountedautomount: no unmounts
```

Running the `adauto.pl` script

You can run the `adauto.pl` script with no command-line options to manually refresh the automount NIS maps on demand. Alternatively, you can manually add the `adauto.reloadtime` configuration parameter to the `/etc/centrifydc/centrifydc.conf` file to control how frequently automount NIS maps are retrieved for the zone. If you manually add this parameter to the configuration files, you can set the value to specify that maps with a time stamp older than the specified number of minutes should be reloaded.

By default, the `adauto.pl` script gets automount NIS maps from the zone to which the local computer is joined. If the maps are not found in the joined zone, the script will attempt to get the maps from its parent zone of the joined zone. Alternatively, you can create the file `/var/centrifydc/kset.automap` and type the common name (CN) of the specific Centrify zone from which you want to load the automount NIS maps.

### Testing the `adauto.pl` script results

After you have configured the `auto.home` and `auto.master` maps, you can test that the `adauto.pl` script is working by entering one of the following commands:

```
/etc/auto.home userid /etc/auto_home userid
```

This command should return the path from the `auto.home` or `auto_home` NIS map stored in Active Directory. For example:

```
/server/home/userid
```

### Restarting the Automount Service

If you make any changes to the NIS maps in Active Directory, you should restart the automount service.

## Distributing Automount Maps

You can create `auto.master` and `auto.home` files as NIS maps in Centrify zones and distribute them using symbolic links to the `adauto.pl` script. In this scenario, you can take advantage of the capability to support executable maps. Depending on your operating system, however, you might be able to take advantage of the Centrify NSS module to automatically mount home directories instead. If your operating system allows you to use the Centrify NSS module, you can add `centrifdc` to the automount line in the `/etc/nsswitch.conf` file.

In most cases, you can use the Centrify NSS module to distribute `auto.home` maps. You cannot use this approach, however, to distribute the `auto.master` map on most operating systems. For the `auto.master` map, your options are typically limited to doing one of the following:

- using NIS.
- using LDAP.
- using a local file.

For information about using LDAP, see “Using the Centrify LDAP proxy service” in the *Administrator’s Guide for Linux and UNIX*. If you use a local file, you can use an `adedit` script to synchronize the `auto.master` map to a local `/etc/auto.master` file. The following example illustrates the steps to synchronize the `auto.master` map to a local `/etc/auto.master` file.

1. Add the File Copy group policy to a Group Policy Object that applies to Centrify-managed computers.
2. Enable the group policy to copy a script similar to the following to the directory `/usr/share/centrifdc/mappers/machine`:

```
#!/bin/sh
# Restart adedit using tclsh \
exec adedit "$@" "$@"
# Bind to an Active Directory domain \
bind -machine domain
# Select a zone context \
select_zone zone
catch {
    select_nis_map auto.master
    set output [open /etc/auto.master w 0644]
    foreach line [gnm] {
        puts $output [regsub ":1" $line ""]
    }
    close $output
}
```

By adding a script similar to this sample script to a GPO, every 90 to 120 minutes the group policy update will execute the script to read the contents of the `auto.master` map in Active Directory and create a local copy of the `/etc/auto.master` file.

You can also use this same approach to synchronize all of the maps stored in Active Directory to the local `/etc` directory. For example:

## Preparing for Agentless Authentication for NIS Clients

```
#!/bin/sh
# Restarts using tclsh \
exec adedit "$0" "$@"
bind -machine [adinfo domain]
slz [adinfo zone]
foreach map [get_nis_maps] {
    if ([regexp "auto*" $map]) {
        slnm $map
        set output [open /etc/$map w 0644]
        foreach line [gnm] {
            puts $output [regsub ":1" $line ""]
        }
        close $output
    }
}
```

## Discontinuing Use of Legacy NIS Servers

If you have existing NIS servers running on your network, you can configure your NIS clients to use the Centrify Network Information Service over time, as needed. Once you have the Network Information Service running, you can also incrementally update the NIS data that's stored in Active Directory using the Access Manager console. Any updates you make are then propagated to all of the `adnisd` servers automatically.

When you are satisfied that you have all of the appropriate NIS information stored in Active Directory and have deployed `adnisd` across the enterprise, as needed, you can then stop any remaining legacy NIS servers and complete the migration to Active Directory for secure, centralized directory service.



Although you can leave the standard NIS servers in place indefinitely, you should plan to migrate all of your data and NIS clients to use the Centrify Network Information Service if you want you to centralize all authentication and directory service in Active Directory. Once you have imported all of the data you need into Active Directory and configured your existing NIS clients to use the Centrify Network Information Service in the appropriate zone, you can decommission any legacy NIS servers and stop any related services.

## Preparing for Agentless Authentication for NIS Clients

This section describes the activities that are specific to preparing your environment to handle agentless authentication and authorization. If you only plan to use Delinea Network Information Service (`adnisd`) to publish network information, such as automount mount points, `netgroup` membership rules, or custom maps, you can skip this chapter.

### Deciding to Use Agentless Authentication

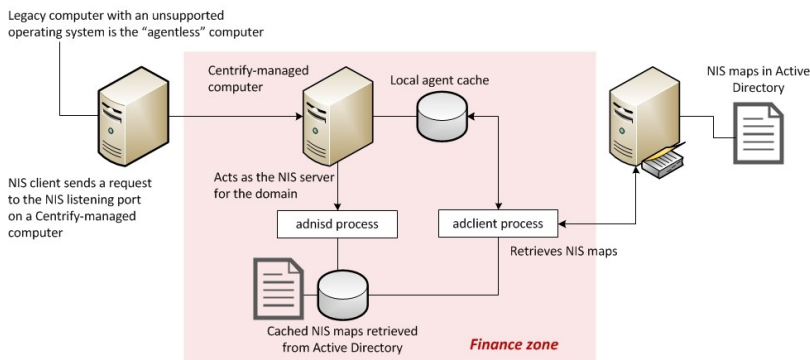
Normally, the `adclient` agent is installed locally on a computer to handle all account authentication and lookup requests that need to be passed to Active Directory. On computers and devices where you cannot install a Delinea

## Preparing for Agentless Authentication for NIS Clients

Agent locally, you may be able to use the Delinea Network Information Service (`adnisd`) to provide agentless authentication.

With agentless authentication, computers that have older or unsupported operating systems that can be, or already are, configured as NIS clients can submit NIS requests to the Delinea Network Information Service. The Delinea Network Information Service can then check its cached Active Directory information to verify whether a user or group has valid credentials and is authorized to log on.

The following figure provides a simplified view of this environment.



In this scenario, the Delinea zone acts as the NIS domain for a group of computers or devices that are configured as NIS clients. Those clients submit requests to the Delinea Network Information Service, `adnisd`, listening on the NIS port.

The Delinea Network Information Service periodically contacts the `adclient` agent to get updated information from Active Directory and generates a set of “maps” that it stores locally. The Delinea Network Information Service can then use the information in these maps to respond to NIS client requests for authentication or other services.

The user and group “maps” are generated automatically based on the users and groups that have profiles currently enabled in the zone. Network information and custom maps can also be published for a zone, but those maps must be manually imported or created. The maps for agentless authentication only require you to add and enable a profile for each Active Directory user and group who should have access to the zone. In this way, the Delinea Network Information Service can be used to service agentless authentication requests from computers or devices where `adclient` itself cannot be installed.

## Planning for Agentless Authentication

In planning a deployment that supports agentless authentication for NIS clients, you should keep in mind that the zone associated with the computer where `adnisd` is installed defines the scope of information available to the NIS clients that the `adnisd` process serves. Each instance of `adnisd` supports one and only one zone, which is equivalent to a single NIS domain. The `adnisd` process can only look up information for the computers, groups, and users that exist in the same zone as the local computer account, and all instances of the `adnisd` in the same zone respond to queries using the same information from Active Directory.

For users and groups to be available for agentless authentication, therefore, they must be enabled for the zone the Delinea Network Information Service serves. In addition, each zone that supports agentless authentication requires an Active Directory attribute for storing the password hash for UNIX-enabled users. The password hash is not created in Active Directory by default, so it must be generated then maintained using a password synchronization

## Preparing for Agentless Authentication for NIS Clients

service installed on all of your domain controllers. The Active Directory attribute that holds the password hash must be accessible to the computers you are using as NIS servers in each zone.



If you install the Delinea Network Information Service on multiple computers, whether in the same zone or across multiple zones, all of these instances are zone-specific peers. There are no master/slave instances.

If you decide you want to use the Delinea Network Information Service to support agentless authentication, you should:

- Identify the zones for which you want to publish information. For example, if you want user and group information broadly available to NIS clients across the network and you have a parent zone, you may want to allow agentless authentication for all of the users in that zone. If you want to strictly control which users can be authenticated to NIS clients, you may want to create a separate agentless-authentication child zone that only contains those users and their groups. For each zone that supports agentless authentication, you must specify the Active Directory attribute for storing the password hash.
- Identify the computers that should service NIS client requests in each zone. You can designate any computer that has the Delinea Agent installed to also act as the Delinea Network Information Server in the zone. Any computer you want to use as the NIS server must have the Delinea Agent for \*NIX installed and must be joined to an Active Directory domain.
- Install a password synchronization service on all of the domain controllers in the joined domain.
- Install and configure the Delinea Network Information Service (`adnisd`) on the selected computers in each zone. On the computers that will act as NIS servers in a zone, you must manually install and start the `adnisd` service. Alternatively, you can modify the startup script on each local computer so that the `adnisd` process starts whenever the local computer is rebooted. You also may want to customize the configuration parameters that control the operation of the `adnisd` process.
- Configure computers and devices as NIS clients that bind to the Delinea Network Information Service on the selected computers in each zone. If any existing NIS servers are running, you will need to reconfigure the NIS clients on the network to use the computer where the Delinea Network Information Service is installed as their NIS server.
- Import and enable the users and groups who need to be authenticated to NIS clients for the zone. You can migrate this information from existing NIS servers or local configuration files by importing `passwd` and `group` NIS maps or local `/etc/passwd` and `/etc/group` files using the **Import from Unix** wizard, or you can manually or programmatically create UNIX profiles for users and groups, as needed. The users and groups must have UNIX profiles stored in Active Directory and enabled for the local computer's zone for the Delinea Network Information Service to generate the maps it needs to service agentless authentication and lookup requests from NIS clients.
- Import and manage any additional NIS maps you want to make available to NIS clients. For example, you can import network maps such as `netgroup` and `automount` NIS maps or create custom maps using the Access Manager console.



Importing existing NIS maps simply provides a mechanism for migrating information to the Active Directory. Once the information is stored in Active Directory, any original NIS maps you imported are no longer used. Instead, the Delinea Network Information Service uses the information stored in Active Directory to automatically generate the maps it needs to service authentication and lookup requests. This local cache of data is updated at a regular interval.

## Selecting a Zone to Use for NIS Authentication

A computer's zone is equivalent to a NIS domain for the Delinea Network Information Service. Each instance of the Delinea Network Information Service supports one and only one zone. All instances of the Delinea Network Information Service in the same zone respond to queries using the same information from Active Directory.

If user information from a zone needs to be available to NIS clients for agentless authentication, the Delinea Network Information Service must be able to access the password hash for zone users. However, because Active Directory does not generate a password hash for users by default, there's no default attribute for storing this information.

To enable the password hash to be stored for users in a zone:

1. Start Access Manager.
2. In the console tree, expand the **Zones** node.
3. Select the zone that will service NIS client requests, right-click, then click **Properties**.

For example, if you want to work with a child zone, `sanfrancisco`, expand the parent zone and Child Zones nodes, select the `sanfrancisco` zone right-click, then click **Properties**.

4. On the General tab, select the **Support agentless client** option.
5. Select the Active Directory attribute to use for storing the password hash.

Depending on the password synchronization service you are using and the Active Directory schema, select one of these attributes:

- **altSecurityIdentities** if you are using the Delinea Password Synchronization program. Do not select this option if you are using a Microsoft password synchronization service.
  - **msSFU30Password** if you are using the Microsoft Windows Services for UNIX Password Synchronization Service. If you are using the Delinea Password Synchronization program, you can choose this attribute if you have the SFU schema installed.
  - **unixUserPassword** if you are using the Microsoft UNIX Identity Management Service and are using the Delinea Password Synchronization program.
6. Verify the default NIS domain name.

By default, the zone name is used as the NIS domain name because this makes it easy to identify the scope of the information available to NIS clients. You can type a different name in the zone properties if you choose.

Whether you use the default name or another name for the NIS domain, you must use the same name when you configure the NIS clients. For more information about configuring NIS clients, see [Configuring NIS clients](#).

7. Click **OK** to save the changes and close the zone Properties.



## Selecting a Computer for NIS Authentication

You can designate any computer in a zone to act as the NIS server for the zone by setting the **Allow this computer to authenticate NIS users** computer property as described in “Adding Support for Agentless Clients” in the *Administrator’s Guide for Linux and UNIX*. For example, expand the Computers node in the zone that will service NIS client requests, select the computer account, right-click to select **Properties**, then click the **Delinea Profile** tab to set this option.

The computer account acting as a NIS server for the zone must be able to access the attribute containing the password hash for agentless authentication to be successful.

Selecting **Allow this computer to authenticate NIS users** adds the computer account to the `zone_nis_servers` Active Directory group. Computer accounts that are placed in the `zone_nis_servers` group are automatically granted permission to read the attribute that stores the password hash for users in the zone.

This property setting enables the computer account to access the password hash so that it can authenticate users in response to NIS client requests. However, you must manually install and start the Delinea Network Information Service on the physical computer before the computer can act as a NIS server.

## Configuring a Password Synchronization Service

The Delinea Network Information Service must be able to retrieve the current password hash for zone users in order for it to respond to agentless authentication requests from NIS clients. Active Directory, however, does not generate a password hash for users by default. This task is handled by the password synchronization service. Therefore, to generate the password hash for zone users, you first need to install a password synchronization service.

You can install the password synchronization service with the Server Suite or separately using a standalone setup program. Once deployed, it ensures the passwords served by the Delinea Network Information Service are always up-to-date. With a password synchronization service, any time users change their Active Directory password, the corresponding password hash in their user profile is updated to reflect the change. Depending on your environment, you can choose to install one of the following:

- Delinea Password Synchronization program
- Microsoft Windows Services for UNIX Password Synchronization Service
- Microsoft Windows UNIX Identity Management Service



Regardless of the password synchronization service you choose to use, the service must be installed on all domain controllers in the Active Directory domain where you are enabling agentless authentication.

## Using Delinea Password Synchronization

You can install the Delinea Password Synchronization program using the Server Suite setup program. Alternatively, you can install Delinea Password Synchronization independent of the the Server Suite using its own setup program. If you install the Delinea Password Synchronization program using the setup program, you can skip this section.

To install the Delinea Password Synchronization program:

## Preparing for Agentless Authentication for NIS Clients

1. Copy the `Centri fyDC_PasswordSync-n.n.n-win64` package to your Active Directory domain controller.
2. Open the `Centri fyDC_PasswordSync-n.n.n-win64` executable or Microsoft software installation (.msi) file to start the setup program.



You can run the setup program interactively or silently if you use the Microsoft software installation (.msi) file. If you are installing silently using the `msiexec` program, you can skip the steps in this section.

3. At the Welcome page, click **Next**.
4. Review the terms of the license agreement. If you accept the license agreement, select **I accept the terms of the license agreement**, then click **Next**.
5. Type your name and company, select who should be able to use this application on the computer, then click **Next**.
6. Select a restart option, then click **Finish**.

Once installed, the Delinea Password Synchronization program will generate the initial password hash when users next change their password, then update the password hash at each password change thereafter. The password hashes are created using DES encryption with a two character salt. If the password hash is stored in the `altSecurityIdentities` attribute, it has a prefix of `cdcPasswordHash`, for example:

```
cdcPasswordHash:vkievQ69VhYKc
```

If the password hash is stored in one of the other supported attributes, it is stored without a prefix.

When a user changes his Active Directory password, the Delinea Password Synchronization program discovers the zones to which that user has access and updates the appropriate attribute that holds the password hash for that user in each zone.



The initial password hash is only generated when the user changes his password. You may want to force users to change their password at the next logon to get the password set at the earliest opportunity. Client authentication requests may fail for users who do not have a password hash available. If the password hash field in the `passwd.byname` or `passwd.byuid` map displays a single exclamation point (!), it indicates that the user's password hash has not been set.

## Using Microsoft Password Synchronization Service

If you choose to use one of the password synchronization services provided by Microsoft instead of the Delinea Password Synchronization program, follow the instructions provided with the software to install the service. In general, you need to do the following to use the Microsoft password synchronization services:

- Set the Windows domain to the domain you joined after installing the Delinea Agent for \*NIX.
- Set the NIS domain name to the zone name you specified when you joined the domain. For example, if you are using the **default** zone, set the NIS domain to **default**. Although you can set the NIS domain name to something other than the zone name when creating or modifying a zone's properties, you must use the zone name for this setting if you use Microsoft password synchronization.
- Set the NIS Server name to the host name of the computer running both the `adcli` and `adnisd` services.

## Configuring the Server Suite NIS server

- Give user accounts access to the zone and NIS domain. If you are using the Microsoft Windows Services for UNIX, select the zone name from the list of NIS domains on the **UNIX Attributes** tab.

The rest of the fields on the UNIX Attributes tab are not used by Server Suite, but you are required to enter information for these fields to enable the NIS domain for the user. Therefore, you should specify a UID, Login shell, Home directory, and Primary group for the user account, then click **OK**.

## Locating Zones for Password Synchronization

Only Active Directory users with a UNIX profile created using the Access Manager console include the attribute (`parentLink`) needed to look up their zone information for password synchronization. You can use the **Orphan Unix data objects** option in the Analyze Wizard to check the forest for accounts missing this attribute setting and attempt to correct the problem.

If the Analysis Results display a **Warning** for the **Orphan Unix data objects** check, you can right-click, then select **Properties** to view additional details. If the profile is missing the `parentLink` attribute, select the warning, right-click, then select **Populate parentLink** attribute to define this attribute for the user.

For more information about troubleshooting issues for the Delinea Network Information Service, see [Troubleshooting and Logging NIS Operations](#). For more information about using the Analyze wizard in the Access Manager console, see “Analyzing information in Active Directory” in the *Administrator’s Guide for Linux and UNIX*.

# Configuring the Server Suite NIS server

This section describes how to install and configure the Centrify Network Information Service (`adnisd`). The `adnisd` process allows a Centrify-managed computer to act as the NIS server for NIS clients in a joined domain. Using `adclient` and `adnisd` together, you can store authentication, authorization and network information in Active Directory, and respond to NIS client requests from computers and devices even where `adclient` cannot be installed.

## Installing the Centrify NIS Server

Whether you want to use the Centrify Network Information Service for agentless authentication, managing network information, or publishing custom maps, you must install and configure `adnisd` on at least one computer in at least one zone before you can begin responding to NIS client requests.

In most cases, `adnisd` is installed as part of a custom installation of the Server Suite or as a separate software package, independent of the installation of `adclient`. The naming convention for the standalone software package is:

```
centrifydc-nis-n.n.n-os-architecture
```

Keep in mind:

- You must install `adnisd` on a computer where `adclient` is also installed.
- The Active Directory domain and zone the local computer has joined defines the NIS domain, and therefore the information available to NIS clients.
- You cannot use `adnisd` to serve NIS maps if your managed computer joined the domain using the `--workstation` option.

## Configuring the Server Suite NIS server

- Using the `--workstation` option adds a computer to the single Auto Zone where user and group profiles are generated automatically. Computers in the Auto Zone cannot be used as NIS servers or NIS clients.
- You can install `adnisd` using any installation program appropriate for the local operating environment, such as RPM, SMIT or YAST.
- If you are upgrading from a previous release of Server Suite and have an earlier version of `adnisd`, stop the existing `adnisd` service and use `install.sh` to remove the old packages before installing the new version of `adclient` and `adnisd`.

The following steps are only an example of how to install `adnisd` locally on a computer. The specific steps required depend on the local operating environment and the installation program you choose.

1. As root on the managed computer, use `adinfo` to verify that `adclient` is installed, and that the local computer is joined to a domain and can connect to Active Directory:

```
su -
Password:

adinfo

Local host name:      magnolia
Joined to domain:    ajax.org
Joined as:           magnolia.ajax.org
Current DC:          ginger.ajax.org
Preferred site:      Default-First-Site-Name
Zone:                ajax.org/Program Data/Centrify/Zones/default
Last password set:   2006-12-28 14:47:57 PST
CentrifyDC mode:     connected
```

2. Copy the package appropriate to the local computer's operating environment, from the Server Suite CD or a download directory, to a local directory.

For example, if the operating environment is Solaris 9 SPARC:

```
cp /tmp/centrifydc-nis-n.n.n-sol8-sparc-local.tgz .
```

3. If the package is a compressed file, unzip and extract its contents. For example, on Solaris:

```
gunzip -d centrifydc-nis-n.n.n-sol8-local.tgz
tar -xf centrifydc-nis-n.n.n-sol8-sparc-local.tar
```

4. Run the appropriate command for installing the package. For example, on Solaris:

```
pkgadd -d CentrifyDC-nis -a admin
```

## Adding IP Addresses from Which to Accept Requests

By default, the Network Information Service accepts only local NIS client requests. To accept requests from any other NIS clients in a network, modify `nisd.securenets` in the `/etc/centrifydc/centrifydc.conf` file to specify the computer subnets from which to accept NIS requests. This parameter configures `adnisd` to filter NIS client requests by IP address. It ignores all other NIS client requests.

For example, to restrict NIS requests to the single trusted subnet 172.68.0.0, add a line similar to the following to `nisd.securenets`:

```
nisd.securenets: 172.68.0.0/255.255.0.0
```

To specify multiple subnets, separate the entries with commas or spaces:

## Configuring the Server Suite NIS server

```
nisd.securenets: 172.68.0.0/255.255.0.0,196.48.0.0/0
```

To accept NIS client requests from any computer, use this:

```
nisd.securenets: 0/0
```

On systems with multiple Ethernet interfaces, `adnisd` configures RPC to the first interface. If an NIS client is trying to communicate on a different interface, `adnisd` will not receive the request.

Before creating sockets, `adnisd` reads the `centrifdc.conf` file to see if an IP address and TCP and UDP ports are specified. If not, it uses localhost and random port numbers assigned by the operating system.

You set the IP address, TCP port and UDP port using the `nisd.net_addr`, `nisd.port.tcp`, and `nisd.port.udp` configuration parameters, respectively in the `centrifdc.conf` file.

For more information, see the *Configuration and Tuning Reference Guide*.

## Starting the `adnisd` Process

After you have specified the subnets from which to accept NIS client requests, you can either manually start the `adnisd` process at the command line, or reboot the local computer. By default, the `adnisd` process starts automatically whenever the computer is rebooted. If you don't want the process started automatically, you should modify the startup script on the local computer to remove `adnisd` from the processes started.



The installer adds the `adnisd` process to a computer's startup script for you. If you are not importing NIS maps right away, you may want to modify the startup script to prevent the `adnisd` process from starting before you are ready to begin servicing client requests.

To start the `adnisd` process at the command line:

1. Verify that `adclient` is running and the local computer is joined to a domain.
2. Verify that RPC is running on the local computer. For example:

```
rpcinfo -p localhost
```

The `adnisd` process requires RPC services. If you restart RPC, you also need to restart the `adnisd` process.

3. Type the appropriate start command. For example, on Red Hat Linux, type:

```
/sbin/service adnisd start
```

On most other platforms, run:

```
/etc/init.d/adnisd start
```

On Solaris 10 or later, the daemon is controlled through the Solaris Service Management Facility. For example:

```
svcadm enable nis/centrifdc-server
```

When the `adnisd` process starts, it connects to Active Directory through `adclient` and does the following:

- Retrieves the current user, group, network and custom information stored in Active Directory for its zone.
- Generates additional maps derived from the retrieved information, such as `netgroup.byuser`, `netgroup.byhost`, `passwd.byuid`, `passwd.byname`, `group.byname`, and `group.bygid`.
- Stores the information retrieved or derived from Active Directory in a local cache of NIS map data.

## Configuring the Server Suite NIS server

After the initial connection, `adnisd` periodically connects to Active Directory through `adclient` to retrieve updated information for its zone. However, `adnisd` always responds to NIS client requests using the data in its local cache so that it can respond to NIS requests even if Active Directory is unavailable.

## Customizing the Update Interval for NIS Maps

By default, every 30 minutes (1800 seconds), `adnisd` uses `adclient` to connect to Active Directory. At the update interval, `adnisd` does the following:

- Checks for network NIS maps explicitly defined in Active Directory to determine whether any records have changed.
- Generates derived maps for any explicitly defined network maps that `adnisd` recognizes. For example, if the `netgroup` map is found in Active Directory, `adnisd` generates the `netgroup.byuser` and `netgroup.byhost` maps.
- Updates the local cache with all changes to the network NIS maps.
- Updates the local cache with changes to the derived maps for user and group information in the zone.



In most cases, updating the local cache of NIS data does not require you to restart any services.

In most organizations, the default update interval is adequate. In a more volatile or stable NIS map environment, reduce or increase the time between updates, as appropriate, by modifying the `nisd.update.interval` parameter in `/etc/centrifydc/centrifydc.conf` to specify a different number of seconds between updates; for example:

```
nisd.update.interval: 900
```

For more information, see the *Configuration and Tuning Reference Guide*.

## Customizing the NIS Maps to PUBLISH

By default, the `adnisd` process retrieves all NIS maps stored in Active Directory at each update interval, updates its local cache as needed, and makes all such data available to its NIS clients. In some cases, you may want to prevent NIS clients from accessing data in specific maps or from looking up information using a specific key.

If you want to customize the list of maps to make available to NIS clients, modify the `nisd.maps` or `nisd.exclude.maps` parameter in `/etc/centrifydc/centrifydc.conf`, or apply a group policy.

- With the `nisd.maps` parameter, you explicitly list the NIS maps, including derived maps, to *include* in the local cache of map data; for example:  

```
nisd.maps: hosts.byname,hosts.byaddr,automount
```
- With the `nisd.exclude.maps` parameter, you list the NIS maps to *exclude* from responses to NIS client requests (typically user and group information). When you specify a map, its derived maps are excluded as well. For example:  

```
nisd.exclude.maps: group passwd
```

For more information, see the *Configuration and Tuning Reference Guide*.

## Configuring the Maximum Number of Map Sets

When `adnisd` receives data for explicitly-defined NIS maps, the data comes from the domain controller selected by the `adclient` process. If the domain controller the `adclient` process has changed - for example, if it is unavailable - the `adclient` process attempts to find another available domain controller.

To ensure the data consistency of the NIS maps retrieved from Active Directory, `adnisd` keeps a separate set of NIS records from each domain controller. This enables `adnisd` to switch between domain controllers efficiently, but uses more space in the local cache.

You can control the maximum number of alternate sets of NIS maps to maintain (default is two) by modifying the `nisd.maps.max` parameter in `/etc/centrifydc/centrifydc.conf`. For example, to keep up to four sets of NIS maps, specify:

```
nisd.maps.max: 4
```

For more information, see the *Configuration and Tuning Reference Guide*.

## Handling Large Active Directory Groups

In most cases, the NIS server cannot send more than 1024 characters of data to NIS clients in response to a query. This limitation can create problems when the NIS client requests information for a large group with a long membership list. By default, the `adnisd` process automatically truncates the list at 1024 characters.

You can configure `adnisd` to split large groups into several groups of conforming size and names using `nisd.largegroup.suffix` and `nisd.largegroup.name.length` in `/etc/centrifydc/centrifydc.conf`.

### Splitting a single large group into multiple new groups

If you specify any value for the `nisd.largegroup.suffix` parameter, `adnisd` splits large groups into multiple new groups automatically, creating a new group whenever a group's data size exceeds 1024-character limit by appending the string you define in `nisd.largegroup.suffix` plus a sequential number.

For example, if you have a large group named `performix-worldwide-corp`, and have defined the suffix string as “-all”, when the `performix-worldwide-corp` group membership is split into multiple groups, the groups are named as follows:

```
performix-worldwide-corp-all1  
.  
.  
performix-worldwide-corp-all n
```

All of the new groups have the same group identifier (GID) as the original group.

### Setting the maximum length of new group names

If the new group names would exceed the maximum length for group names on a platform, use the `nisd.largegroup.name.length` parameter. If you do this, `adnisd` truncates the original group name so as not to exceed the maximum name length.

For the example above, if you set a maximum name length of 14, the split groups are named:

```
performix-all1  
.  
.  
permi-all10
```

## Configuring NIS Clients

```
...  
perform-a11100
```

All of the new groups have the same group identifier (GID) as the original group.

For more information, see the *Configuration and Tuning Reference Guide*.

## Making the NIS Server Available

After you install and configure `adnisd` on a computer, you must configure other computers or devices on the network to use the computer running `adnisd` for NIS client requests.

In general, configuring NIS clients to use the Centrify Network Information Service involves:

- Stopping any existing legacy NIS server processes.
- Modifying the NIS client's configuration file to identify the zone and computer name of the computer where the `adnisd` process is installed.
- Sending a `bind` request from the NIS client to the new Centrify NIS server.

For more information, see [Configuring NIS clients](#).

## Configuring NIS Clients

This section describes how to configure NIS clients to receive authentication, authorization, and network information through the Delinea Network Information Service.

### Specifying the Server for NIS Clients to Use

After you install and configure `adnisd` on a computer, you must configure other computers or devices to send their NIS lookup requests to the computer running `adnisd`. The specific steps for configuring the NIS client are slightly different in different operating environments. In general, configuring NIS clients involves:

- Stopping the connection to any existing NIS server.
- Identifying the zone and computer name of the computer where `adnisd` is installed in the client's NIS configuration file.
- Binding to the new Delinea NIS server.
- Restarting services that use NIS, or rebooting the computer.

For information about configuring the NIS client in different operating environments, see the appropriate section below.



The client configuration instructions assume that you are using the zone name as the NIS domain name. If not, substitute the NIS domain name you specified when you created the zone where applicable. For more information about configuring NIS clients on any specific platform and OS version, consult the documentation for that platform.



### Configuring NIS Clients on Linux

To configure the NIS client on a Linux computer:

1. Stop any running NIS service and remove all files from the `/var/yp/binding` directory. For example, run the following commands:

```
/sbin/service ypbind stop  
rm -rf /var/yp/binding/*
```

2. Set the NIS domain name for the client to the zone name or NIS domain name of the computer where the `adnsd` process is running.

```
domainname zone_name
```

For example, if you have installed `adnsd` on a computer in the `corpHQ` zone:

```
domainname corpHQ
```

3. Edit the NIS configuration file, `/etc/yp.conf`, to specify the `Delinea` zone and the name of the computer where `adnsd` is installed.

```
domain zonename server hostname
```

For example, add a line similar to this to `/etc/yp.conf`:

```
domain corpHQ server localhost
```

If your NIS clients are configured for broadcast discovery, this step may not be necessary.

4. Start the `ypbind` service.

On Red Hat Linux, run:

```
/sbin/service ypbind start
```

On Debian 3.1, run the `nis` script (controlled using the file `/etc/default/nis`). By default, the script starts the NIS client, `ypbind`. For example, run the following command:

```
/etc/init.d/nis start
```

On SuSE Linux 9.3 Professional, run:

```
/etc/init.d/ypbind start
```

5. Modify the `passwd`, `group`, and `shadow` lines in `/etc/nsswitch.conf` file to use `compat` as the source:

```
passwd: compat
```

```
group: compat
```

```
shadow: compat
```

6. Restart services that rely on the NIS domain, or reboot the computer to restart all services. The most common services to restart are `autofs`, `NSCD`, `cron` and `sendmail`.

### Configuring NIS Clients on Solaris

To configure the NIS client on a Solaris computer:

## Configuring NIS Clients

1. Stop any running NIS service and remove all files from the `/var/yp/binding` directory. For example, run the following commands on Solaris 8 or 9:

```
kill ypbind
rm -rf /var/yp/binding/*
```

On Solaris 10, stop the service by running:

```
svcadm disable network/nis/client
```

2. Set the NIS domain name for the client to the zone name of the computer where `adnsd` is running.

```
domainname zone_name
```

For example, if you have installed `adnsd` on a computer in the `corpHQ` zone:

```
domainname corpHQ
```

3. Run the `ypinit -c` command and enter the name of the computer where `adnsd` is installed.

This step is not required if you use the `broadcast` option to locate the server when you run the `ypbind` command. You must use `ypinit`, however, if your network topology would prevent a broadcast from reaching the desired servers. For example, if the router does not transmit broadcasts across subnets, use the `ypinit -c` command to specify a server on a different subnet.

Start the `ypbind` service. On most versions of Solaris, run:

```
/usr/lib/netshvc/yp/ypbind
```

If you are using the `broadcast` option to locate the server, start the service with that option. For example:

```
/usr/lib/netshvc/yp/ypbind -broadcast
```

On Solaris 10, run:

```
svcadm enable network/nis/client
```

Modify the `passwd`, `group`, and `shadow` lines in `/etc/nsswitch.conf` file to use `compat` as the source:

```
passwd: compat
group: compat
shadow: compat
```

Restart services that rely on the NIS domain or reboot the computer to restart all services. The most common services to restart are `autofs`, `NSCD`, `cron` and `sendmail`.

## Configuring NIS Clients on HP-UX

To configure the NIS client on an HP-UX computer:

1. Stop any running NIS service and remove all files in the `/var/yp/binding` directory. For example, run the following commands:

```
/sbin/init.d/nis.client stop
rm -rf /var/yp/binding/*
```

2. Edit the NIS configuration file, `/etc/rc.config.d/namesvrs`, to set the `NIS_CLIENT` to 1 and the `NIS_DOMAIN` to the name of the Delinea zone. For example:

## Configuring NIS Clients

```
NIS_CLIENT=1
```

```
NIS_DOMAIN="zone-name"
```

3. Add the `-ypset` option to the `YPBIND_OPTIONS` variable and set the `YPSET_ADDR` variable to the IP address of the computer where `adnsd` is installed. For example:

```
YPBIND_OPTIONS="-ypset"
```

```
YPSET_ADDR="15.13.115.168"
```

This step is not required if you want to use the `broadcast` option to locate the server when you run the `ypbind` command.

4. Set the NIS domain name for the client to the zone name of the computer where the `adnsd` process is running.

```
domainname zone_name
```

5. Start the `ypbind` service. On HP-UX, you can start the service by running:

```
/sbin/init.d/nis.client start
```

6. Modify the `passwd`, `group`, and `shadow` lines in `/etc/nsswitch.conf` file to use `compat` as the source:

```
passwd: compat
```

```
group: compat
```

```
shadow: compat
```

7. Restart services that rely on the NIS domain or reboot the computer to restart all services. The most common services to restart are `autofs`, `pwgrd`, `cron` and `sendmail`.

## Configuring NIS Clients on AIX

To configure the NIS client on an AIX computer:

1. Stop any running NIS service and remove all files from the `/var/yp/binding` directory. For example, run:

```
stopsrc -s ypbind
```

If the computer is not already a NIS client, you can use the System Management Interface Tool (`smit`) and the `mkclient` command to add `adnsd` to the computer.

2. Open the `/etc/rc.nfs` file and verify that the `startsrc` command is configured to start the `ypbind` daemon:

```
if [ -x /usr/etc/ypbind ]; then
```

```
startsrc -s ypbind
```

```
fi
```

3. Set the client's NIS domain name to the zone name of the computer where `adnsd` is running. For example:

```
domainname zone_name
```

4. Start the `ypbind` service:

```
startsrc -s ypbind
```

5. Restart services that rely on the NIS domain or reboot the computer to restart all services. The most common services to restart are `autofs`, `NSCD`, `cron` and `sendmail`.



Theadnisd service is not supported in a workload partitioning (WPAR) environment (Ref: CS-30588c).

## Verifying the Client Configuration

Run the `domainname` command to verify that the client is configured to use the appropriate Delinea zone or NIS domain name. For example, if you have configured a computer to service NIS requests for the `sanfrancisco` zone and are using the zone name as the NIS domain name:

```
domainname
sanfrancisco
```

To test that the client can connect to the Delinea Network Information Service, run one or more NIS client request commands; for example:

```
ypwhich
ypwhich -m
ypcat -k mapname
```

## Checking the Derived passwd and Group Maps

On a computer you have configured as an NIS client, verify that the NIS maps required for agentless authentication are available by running the following command:

```
ypwhich -m
```

At a minimum, you should see the `passwd.*` and `group.*` map names, followed by the name of the computer you are using as the NIS server. For example, if the computer running `adclient` and `adnisd` is `iceberg-hpux`, you should see output similar to this:

```
passwd.byuid iceberg-hpux
passwd.byname iceberg-hpux
group.byname iceberg-hpux
group.bygid iceberg-hpux
```

These `passwd.*` and `group.*` maps are automatically generated based on the information stored in Active Directory for the zone, including all Active Directory users and groups granted access to the zone. You can view information from any of these maps using a command like `ypcat passwd.byname`. The output displayed should look similar this:

```
paul:xq2UvSkNngA:10000:10000:paul:/home/paul:/bin/bash
mlopez:!:10002:10000:Marco Lopez:/home/mlopez:/bin/bash
jsmith:!:10001:10000:John Smith:/home/jsmith:/bin/bash
```

In this example, the user `paul` has a password hash, but users `mlopez` and `jsmith` do not.

If a user account is new, disabled, locked, requires a password change, or is not enabled for a zone, the Delinea NIS server sets the user's hash field to "!"



On some platforms, you may see `ABCD!efgh12345$67890` as the password hash for users who need to set their password.

## Importing and Managing NIS Maps

This section describes how to import, create and manage NIS maps and map entries using the Access Manager console.



You can also use ADSI Edit, ADEdit, custom scripts or other tools to add, modify and remove NIS maps and map entries. To import NIS maps, however, you must use the Access Manager console.

### Importing and Creating User and Group Profiles

If you want to make user and group information available to NIS clients, whether for agentless authentication or in response to other lookup requests, you must first make sure the appropriate users and groups have zone profiles and role assignments defined in the zone. The zone information is used for automatic generation of the maps `passwd.byname`, `passwd.byuid`, `group.byname`, and `group.bygid`. If you disable a user profile in the zone, the user's information cannot be retrieved or published in response to NIS client requests, or used to authenticate the user's identity.

You can import existing user and group information directly from existing NIS servers and domains or from properly formatted text files, such as `local/etc/passwd` and `/etc/group` files, using the **Import from UNIX** wizard, or you can create new profiles for Active Directory users using the Access Manager console.

Once the appropriate user and group profiles have been added to the zone you are using as a NIS domain, the information is available to NIS clients unless you explicitly restrict the publication of this information.



For information about restricting the maps published, see *Customizing the NIS maps to publish*. For information about importing or creating user and group profiles in a zone, see the *Administrator's Guide for Linux and UNIX*.

### Publishing Network or Custom Information

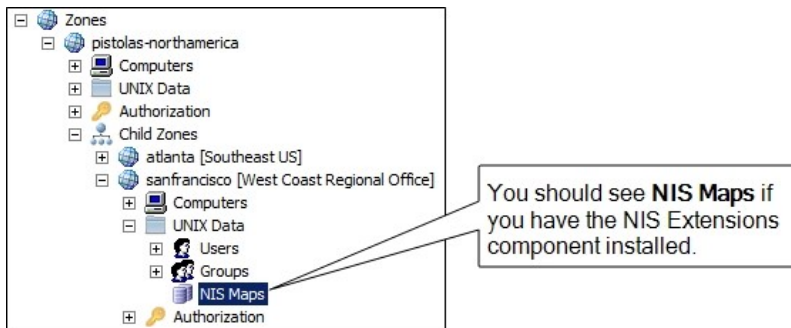
In addition to user and group information, `adnisd` can publish network information or make custom information available to NIS clients. For example, you can import information from standard NIS maps such as `automount`, `netgroup`, and `automaster`, if these maps exist in your environment. Importing network information or creating custom maps, however, requires you to have the NIS Extensions.



NIS Extensions are installed by default when you run the setup program. If you did not select this option, rerun the setup program and select **NIS Extensions** from the list of Access Manager Administration components.

If you have the NIS Extensions installed, you should see the NIS Maps node under each zone. For example, if you are using hierarchical zones, you can see NIS Maps under the UNIX Data node for the parent or child zone you select:

## Importing and Managing NIS Maps



## Importing Network NIS Maps

To use Access Manager to import a standard network NIS map into Active Directory:

1. Open Access Manager.
2. In the console tree, navigate to the specific zone into which you want to import NIS maps.
3. Expand the console tree to display NIS Maps.
4. Select NIS Maps, right-click, then click **Import Maps**.
5. Select whether you want to connect to the NIS server and domain or import the information from a text file, then click **Next**.
  - If you are importing maps directly from an existing NIS server, type the name of the **NIS domain** and **NIS server**. Using this option requires network connectivity to the NIS server from the Window computer you are using.
  - If you are importing a map from a text file, click **Browse** to navigate to the map file you want to import. If you cannot connect directly to the NIS server, you should export the NIS database to a file; then import the information using this option.
6. Select the NIS maps to import if you are importing directly from an existing NIS server, or type a map name and define the file format if importing from a file, then click **Next**. The Import Maps wizard does not validate the information to be imported. If the map has invalid entries, they are imported as-is.

If you importing from a text file:

- Type a **Map name** that describes the type of map being imported. In most cases, you should use the base name that identifies the configuration file used to generate the NIS database. For example, use hosts to identify the map generated from the /etc/hosts file.
- Type the **Field separator** character used to separate fields in the map file.
- Type the column number that defines the start of the **Key field**.
- Specify any additional options as appropriate for the file you are importing. For example, select **Comments are included in the file after** and type the character used to designate comments if the file includes comments.

For Access Manager to correctly interpret the map file, you need to provide accurate information about the file format, such as the type of separator used between fields.

## Importing and Managing NIS Maps

Because the Centrify NIS server does not include comments in response to service requests, you must save the map to a text file and import from that file to retrieve comments contained in NIS maps.

7. When the import is complete, click **Finish**.
8. After importing NIS maps, restart the `adnisd` service.

## Creating New NIS Maps in Active Directory

If you cannot import network information from existing NIS maps, you can create new maps by adding the appropriate information directly to Active Directory using Access Manager. Once you add the information to Active Directory, `adnisd` can use the information to automatically generate a local cache of the map data and make the information in those generated maps available to NIS clients.



If you are creating NIS maps manually, keep in mind that the Network Information Service can return a maximum of 1024 characters of data in response to a query from any NIS map, so make sure all NIS map entries have less than 1024 characters of data.

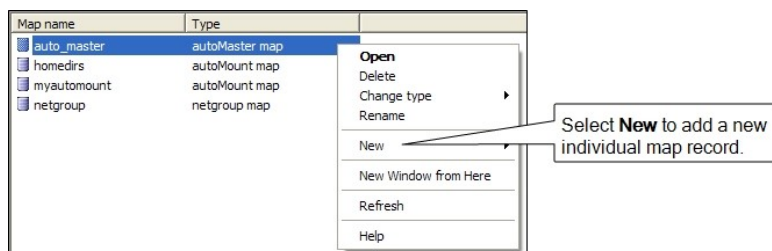
To create a new network NIS map in Active Directory

1. Open Access Manager.
2. Navigate to the specific zone for which you want to create maps.
3. Expand the console tree to display NIS Maps.
4. Select NIS Maps, right-click, then click **New** and select the type of map you want to create.

For most map types, you can only use the recognized map name for the new map. Recognized map names enable you to use derived maps to retrieve information using different keys. If you are creating a new **Automaster** map, you must choose either `auto_master` or `auto.master` as the map name to retrieve the names of the automount maps.

If you select the **Generic Map** option, you can create a custom NIS map for any key/value pairs that you want to make available to NIS clients. For more information, see [Creating generic custom maps](#).

5. Select the new empty map, right-click, then click **New > Map Entry** or **New > netgroup** to add a new individual map record.



The file format and the specific fields used in individual map records depend on the type of map you are working with.

6. Type the appropriate information for the fields listed, then click **OK** to save a record in the new map.

## Importing and Managing NIS Maps

For more information about the fields required in any NIS map, see the man page for the type of map you are creating. For example, see the man page for `netgroup` to see detailed information about required and optional fields and the format of `netgroup` maps.

You can use Active Directory groups in `netgroup` records. Using Active Directory groups in `netgroup` records enables dynamic changes to user and computer pairings based on their Active Directory group membership. If you have existing processes for adding and removing users and computers in Active Directory groups, you can leverage those processes in `netgroup` records.

## Creating Maps for Common Network Services

Centrify uses explicitly-defined NIS maps to generate derived maps automatically. Once a recognized base map is imported or created manually in Active Directory, the agent generates and stores its derived maps so that information can be retrieved searching on different keys.



In most cases, you can import recognized base maps directly from an existing NIS server and domain or from generated text files (for example, files created using the `niscat` command). Alternatively, you can create the base maps manually using the corresponding map type in Access Manager.

The following table describes the recognized base maps and their derived maps.

### aliases

The `aliases` map is the abbreviated name for the `mail.aliases` map. The derived maps are `mail.aliases` and `mail.byaddr`. In most cases, the NIS map is created from the `/etc/aliases` or `/etc/mail/aliases` file. A typical line looks like this:

```
alias: address1 [address2 addressn...] # comment
```

For example:

```
acme: amy.adams@acme.com bill.byrnes@acme.com
widgetco: aaron@widgetco.com,...,zuza@widgetco.com
```

source	<pre>/etc/aliases or /etc/mail/aliases widgetco: aaron@widgetco,...,zuza@widgetco.com</pre>
NIS entry	<pre>mail.aliases Key: widgetco Value: aaron@widgetco,...,zuza@widgetco.com</pre>
	<pre>mail.byaddr Key: aaron@widgetco,...,zuza@widgetco.com Value: widgetco ... Key: zuza@widgetco.com Value: widgetco</pre>
AD entry	<pre>Key: widgetco Value: aaron@widgetco,...,zuza@widgetco.com Comment:</pre>

For the `mail.alias` map, the entries are defined like this:



## Importing and Managing NIS Maps

- Key is the alias name: acme
- Value is the list of addresses for the alias: amy.adams@acme.combill.byrnes@acme.com

For the mail.byaddr map, the entries are defined like this:

- Key is an address: amy.adams@acme.com
- Value is the corresponding alias: acme

If you create an aliases map in Active Directory, you must include the key as part of the value. For example:

- Key: acme
- Value: acme: someone@acme.com
- Comment: someone@acme.com is the address

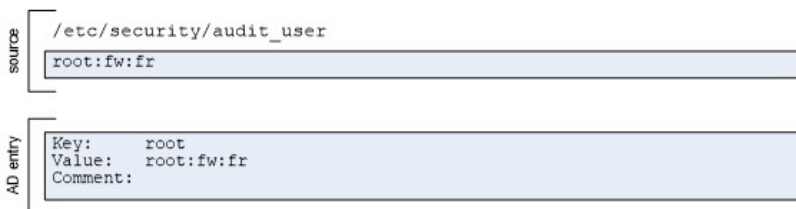
### audit\_user

In most cases, the audit\_user map is created from the /etc/security/audit\_user file. A typical line looks like this:

```
user_name:always_audit_flags:never_audit_flags
```

For example:

```
root:lo:no  
wily:lo,am:io,c1  
kris:lo,ex,+fc,-fr,-fa:io,c1
```



For the audit\_user map, entries are defined like this:

- Key is the user name: root
- Value takes the following format: user\_name:always\_audit\_flags:never\_audit\_flags

If you create an audit\_user map in Active Directory, you must include the key as part of the value. For example:

- Key: root
- Value: root:lo:no

This map is only applicable for Solaris.

### auth\_attr

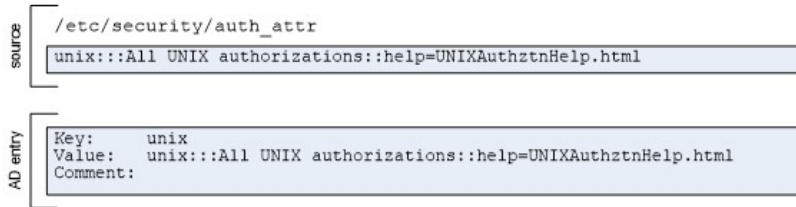
In most cases, the auth\_attr map is created from the /etc/security/auth\_attr file. A typical line looks like this:

```
name:res1:res2:short_description:long_description:attr
```

For example:

## Importing and Managing NIS Maps

```
solaris.::All Solaris Authorization::help=SolarisAuth.html
solaris.user.manage.::Manage Users::help=ManageUsers.html
```



If you create an `auth_attr` map in Active Directory, you must include the key as part of the value. For example:

- Key: `solaris.`
- Value: `solaris.::AllSolarisAuthorizations::attribute`
- Comment: This map provides authorization attributes for `solaris.`

This map is only applicable for Solaris.

## bootparams

In most cases, the `bootparams` map is created from the `/etc/bootparams` file. A typical line looks like this:

```
client_name key=value:[key=value:...]
```

For example:

```
client root=sr04:/export/client/root domain=nyc.test
engr1 root=smoketest:/export/engr1/root rootopts=:vers=2
```



If you create a `bootparams` map in Active Directory, the value must consist of key and value pairs. For example:

- Key: `client`
- Value: `root=sr04:/export/client/root domain=nyc.test`
- Comment: The value consists of key=value pairs separated by colons (:).

This map is only applicable for Solaris.

## ethers

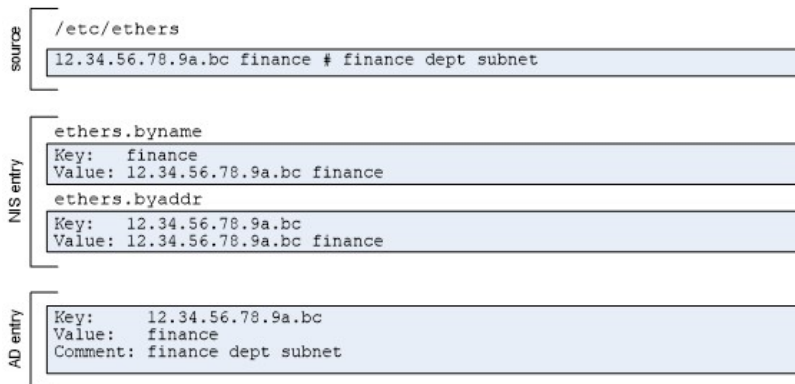
The `ethers` map is the abbreviated name for the `ethers.by_name` map. The derived maps are `ethers.byname` and `ethers.byaddr`. In most cases, the NIS map is created from the file `/etc/ethers` file. A typical line looks like this:

```
ethernet_address host_name
```

For example:

## Importing and Managing NIS Maps

```
52:ef:75:72:4e:c8 rhe19
31:ee:c5:72:4e:18 finance
```



For the `ethers.byname` map, entries are defined like this:

- Key is the host name: `rhe19`
- Value is the ethernet address for the host name: `52:ef:75:72:4e:c8`

For the `ethers.byaddr` map, entries are defined like this:

- Key is an address: `52:ef:75:72:4e:c8`
- Value is the host name: `rhe19`

If you create an `ethers` map in Active Directory, you must include the key as part of the value. For example:

- Key: `rhe19`
- Value: `52:ef:75:72:4e:c8 rhe19`
- Comment: The host name for `52:ef:75:72:4e:c8` is `rhe19`

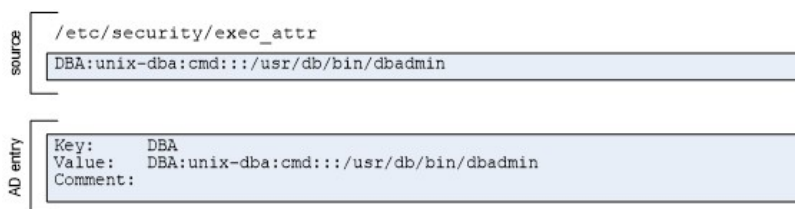
## exec\_attr

In most cases, the `exec_attr` map is created from the `/etc/security/exec_attr` file. A typical line looks like this:

```
name:policy:type:res1:res2:id:attr
```

For example:

```
Application Server Management:suser:cmd:::/usr/bin/admin:
DBA:unix-dba:cmd:::/usr/db/bin/dbadmin:
dbuser:unix-dbuser:cmd:RO::/usr/sbin/db/openssl
```



If you create an `exec_attr` map in Active Directory, you must include the key as part of the value. For example:

## Importing and Managing NIS Maps

- Key: Application Server Management
- Value: execution profile name and properties followed by attributes defined as key and value pairs for the profile:

```
Application Server Management:suser:cmd::: \  
/usr/appserver/bin/admin:
```

This map is only applicable for Solaris.

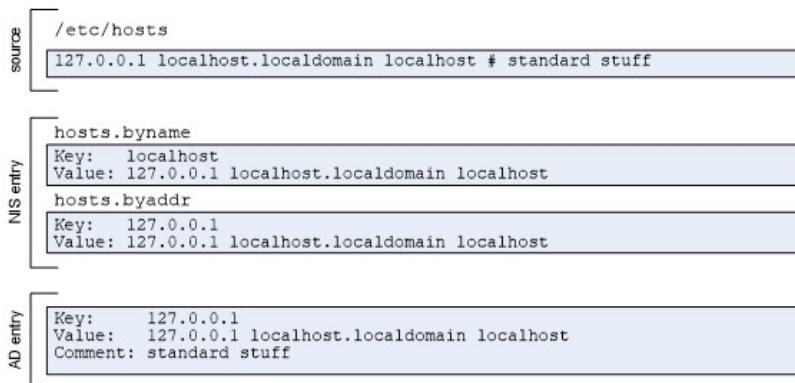
## hosts

The hosts map is the abbreviated name for the hosts.byname map. The derived maps are hosts.byname and hosts.byaddr. In most cases, the NIS map is created from the /etc/hosts file. A typical line looks like this:

```
host_ip_address host_name [alias,...] # comment
```

For example:

```
127.0.0.1 localhost.localdomain localhost  
192.168.22.1 arcade.cendura.net arcade arc1 # clustername
```



For the hosts.byname map, entries are defined like this:

- Key is the host name: `localhost`
- Value is the IP address and any aliases defined for the host: `127.0.0.1 localhost.localdomain localhost`

For the hosts.byaddr map, entries are defined like this:

- Key is an address: `127.0.0.1`
- Value is the IP address and any aliases defined for the host: `127.0.0.1 localhost.localdomain localhost`

If you create a hosts map in Active Directory, you must include the key as part of the value. For example:

- Key: `127.0.0.1`
- Value: IP address and any aliases defined for the host:  
`127.0.0.1 localhost.localdomain localhost`
- Comment: The value includes both the host name and IP

### netgroup

The netgroup map defines a hierarchy of netgroupgroups and members. The netgroupmap controls access by user name, host name, or NIS domain name. The derived maps are netgnetgroup.byhostroup.byhost and netgroup.byuser. In most cases, the NIS map is created from the /etc/netgroup file. A typical line looks like this:

```
netgroup_name (host,user,NIS_domain)[,netgroup]...
```

The keys in a netgroupmap are the names of each netgroup. The values in a netgroupmap are one or more space-separated elements. An element can be:

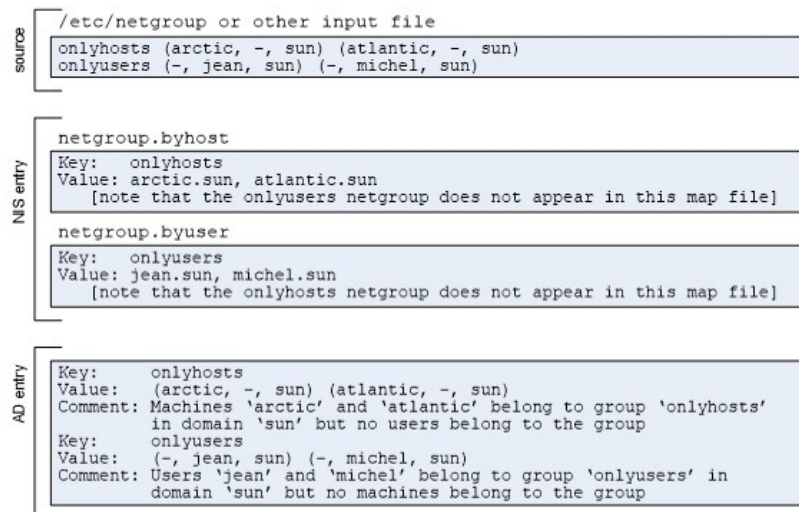
- a set of three comma-separated components.
- a netgroupname.

When specifying an element as a set of three components, you can omit any component to allow any value for that component or specify the special character dash (-) to eliminate a component as a valid value.

The netgroup.byhost map uses the host name as the key and the value is the list of all netgroups that contain the key host somewhere in the hierarchy.

The netgroup.byuser map uses the user name as the key and the value is the list of all netgroups that contain the key user somewhere in the hierarchy.

If you create a netgroupmap in Active Directory, you must not include the key as part of the value. To illustrate, the following example has entries for two netgroups—onlyhosts and onlyusers—and how the groups become key and value entries in the derived NIS maps.



### netmasks

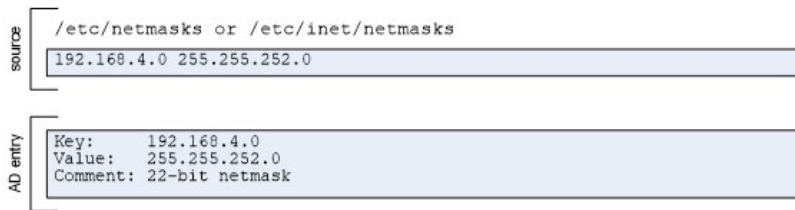
In most cases, the netmasks map is created from the /etc/inet/netmasks or /etc/netmasks file. A typical line looks like this:

```
IP_addressnetmask # comment
```

For example

## Importing and Managing NIS Maps

```
192.168.4.0 255.255.252.0
192.168.4.1 255.255.255.0
```



If you create a netmasks map in Active Directory, you must not include the key as part of the value. For example:

- Key: 192.168.4.0
- Value: 255.255.252.0
- Comment: This is a 22-bit netmask.

This map is only applicable for Solaris.

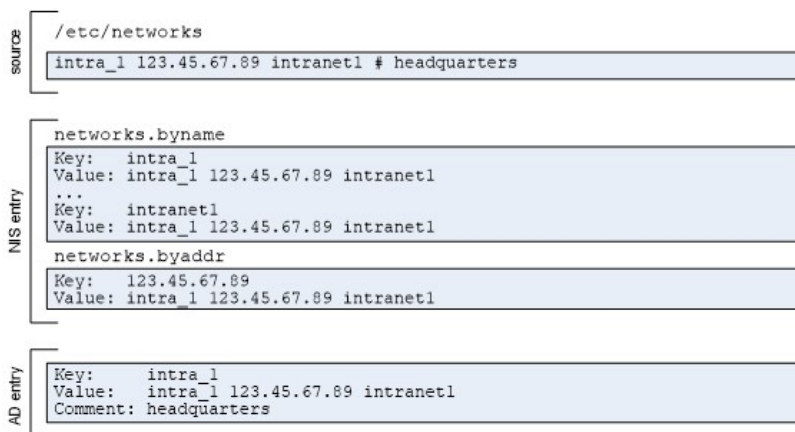
## networks

The networks map is the abbreviated name for the networks.byaddr map. The derived maps are networks.byname and networks.byaddr. In most cases, the networks map is created from the /etc/networks file. A typical line looks like this:

```
network_name network_address [alias1,...] # comment
```

For example:

```
arpa 10 arpanet
intra_1 123.45.67.89 intranet # headquarters
sf_site 171.22.0.0 sf1 # san francisco satellite
```



For the networks.byname map, entries are defined like this:

- Key is the network name: intranet
- Value is the network address and any aliases defined for the network: intranet 171.22.0.0 intra

For the networks.byaddr map, entries are defined like this:

## Importing and Managing NIS Maps

- Key is the network address: 171.22.0.0
- Value is the network name and any aliases defined for the network: intranet 171.22.0.0 intra

If you create a networks map in Active Directory, you must include the key as part of the value. For example:

- Key: intranet
- Value: intranet 171.22.0.0 intra
- Comment: The value includes the network name and address

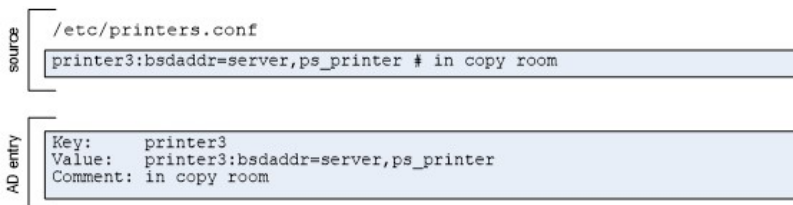
## printers

In most cases, the printers map is created from the `/etc/printers.conf` file. A typical line looks like this:

```
destination_name key=value[,key=value,...] # comment
```

For example:

```
buildx:paddr=buildx.acme.com,105004,1,sys,lp,buildxsp1,1:  
printer3:bsdaddr=server,ps_printer # in copy room
```



If you create a printers map in Active Directory, you must include the key as part of the value. For example:

- Key: printer3
- Value: printer name followed by key and value pairs for the printer properties:  
`printer3:bsdaddr=server,ps_printer`
- Comment: in copy room

This map is only applicable for Solaris.

## prof\_attr

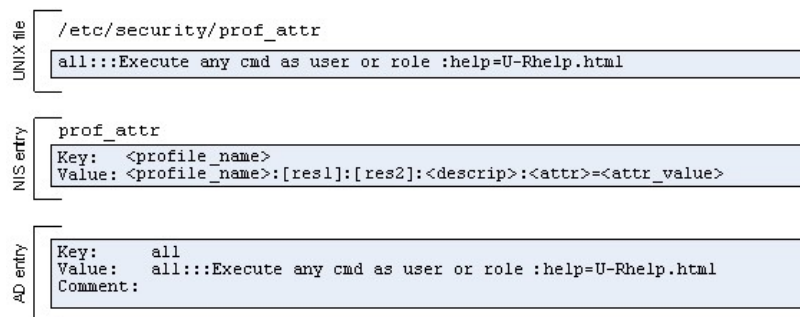
In most cases, the `prof_attr` map is created from the `/etc/security/prof_attr` file. A typical line looks like this:

```
profile_name:res1:res2,description:attr
```

For example:

```
all:::Execute any command as the user:help=AllRights.html  
guest:R0::Allow read-only:audit-flags=all:project=web
```

## Importing and Managing NIS Maps



If you create a `prof_attr` map in Active Directory, you must include the key as part of the value. For example:

- **Key:** `all`
- **Value:** profile name and properties followed by attributes defined as key and value pairs for the profile:  
`all::Execute any cmd as user or role:help=All.html`

This map is only applicable for Solaris.

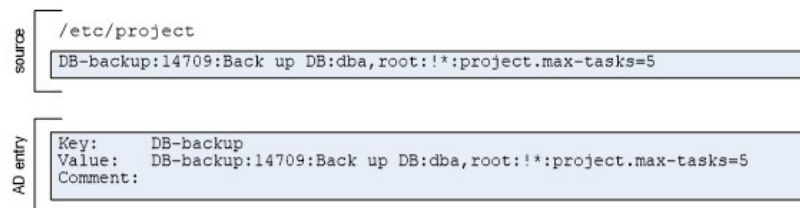
### project

In most cases, the `project` map is created from the `/etc/project` file. A typical line looks like this:

```
project_name:projectid:comment:user_list:group_list:attr
```

For example:

```
DB-backup:14709:Back up DB:dba,root:!*:project.max-tasks=5  
web:101:web services deployment:root:as-team: \\ task.maxltps=  
(privileged,101,signal=SIGTERM)
```



If you create a `project` map in Active Directory, you must include the key as part of the value. For example:

- **Key:** `DB-backup`
- **Value:** project name and properties followed by attributes defined as key and value pairs:  
`DB-backup:14709:Back up DB:dba,root:!*: \\ project.maxtasks=5`

This map is only applicable for Solaris.

### protocols

The `protocols` map is the abbreviated name for the `protocols.bynumber` map. The derived maps are `protocols.byname` and `protocols.bynumber`. In most cases, the `protocols` map is created from the `/etc/protocols` file. A typical line looks like this:

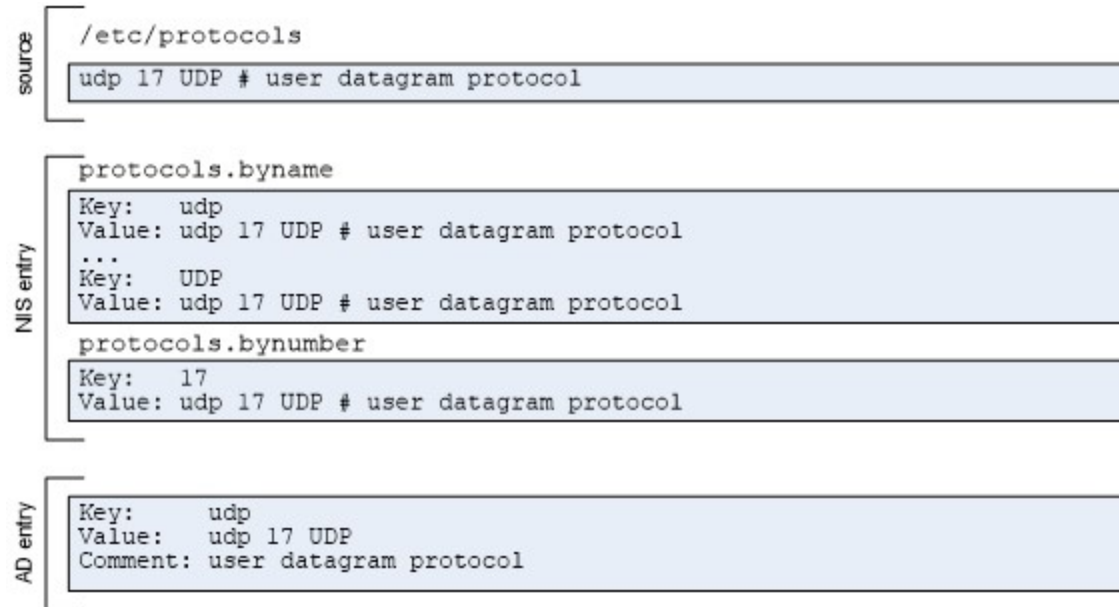
```
protocol number alias # comment
```



## Importing and Managing NIS Maps

For example:

```
ip 0 IP # internet protocol, pseudo protocol number
udp 17 UDP # user datagram protocol
```



For the `protocols.byname` map, entries are defined like this:

- Key is the protocol name: `udp`
- Value is the protocol name, number, and any aliases defined for the protocol: `udp 17 UDP`

For the `protocols.bynumber` map, entries are defined like this:

- Key is the protocol number: `17`
- Value is the protocol name, number, and any aliases defined for the protocol: `udp 17 UDP`

If you create a `protocols` map in Active Directory, you must include the key as part of the value. For example:

- Key: `udp`
- Value: `udp 17 UDP`
- Comment: `user datagram protocol`

## rpc

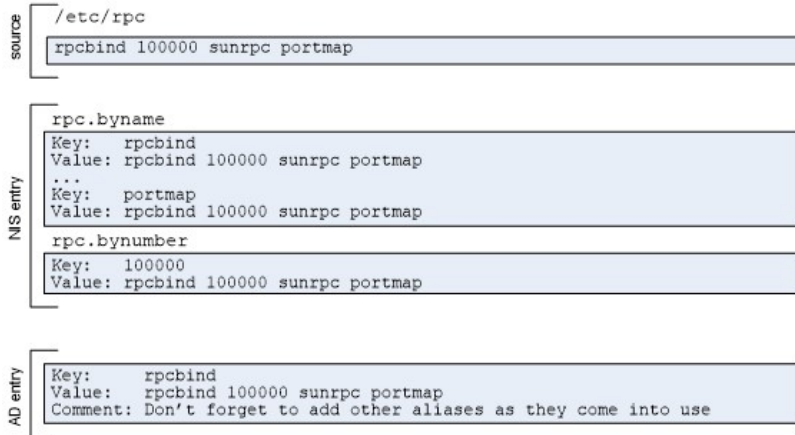
The `rpc` map is the abbreviated name for the `rpc.bynumber` map. The derived maps are `rpc.byname` and `rpc.bynumber`. In most cases, the `rpc` map is created from the `/etc/rpc` file. A typical line looks like this:

```
rpc_name port_number alias1 alias2 ... # comment
```

For example:

```
portmapper 100000 portmap sunrpc
rpcbind 100001
```

## Importing and Managing NIS Maps



For the `rpc.byname` map, entries are defined like this:

- Key is the rpc name or alias, so there would be separate entries for: `portmapper`, `portmap`, `sunrpc`, and `rpcbind`.
- Value for each of the `portmapper`, `portmap`, and `sunrpc` key entries would be the same: `portmapper 100000 portmap sunrpc`

For the `protocols.bynumber` map, entries are defined like this:

- Key is the rpc number: `100000`
- Value is the rpc name, number, and aliases: `portmapper 100000 portmap sunrpc`

If you create a `rpc` map in Active Directory, you must include the key as part of the value. For example:

- Key: `portmapper`
- Value: `portmapper 100000 portmap sunrpc`
- Comment: `portmap and sunrpc are aliases for portmapper`

## services

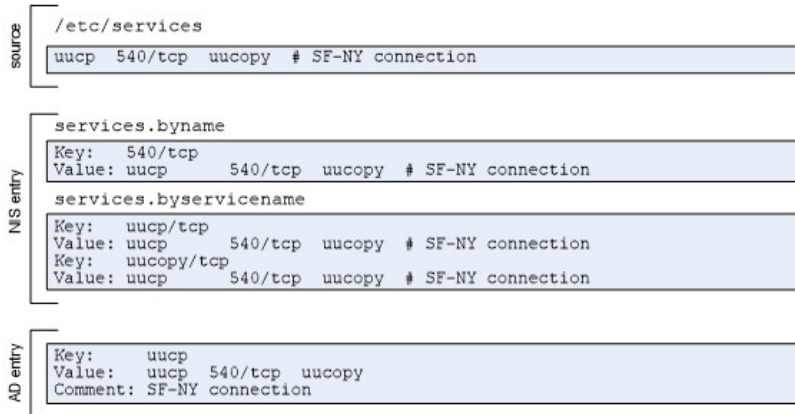
The `services` map is the abbreviated name for the `services.byname` map. The derived maps are `services.byname` and `services.byservicename`. In most cases, the `services` map is created from the `/etc/services` file. A typical line looks like this:

```
service port/protocol alias1 alias2 ... # comment
```

For example:

```
uucp 540/tcp uucopy # this entry is for uucp
```

## Importing and Managing NIS Maps



For the `services.byname` map, entries are defined like this:

- Key is the service name or alias, so there would be separate entries for: `uucp` and `uucopy`.
- Value for each of the `uucp` and `sunrpc` key entries would be the same: `uucp 540/tcp uucopy`

For the `service.byservicename` map, entries are defined like this:

- Key is the port number and protocol: `540/tcp`
- Value contains the same set of fields: `uucp 540/tcp uucopy`

If you create a `services` map in Active Directory, you must include the key as part of the value. For example:

- Key: `uucp`
- Value: `uucp 540/tcp uucopy`
- Comment: `uucopy is an alias for uucp`

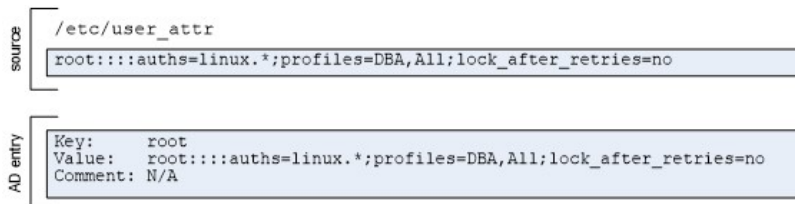
### user\_attr

In most cases, the `user_attr` map is created from the `/etc/user_attr` file. A typical line looks like this:

```
user:qualifier:res1:res2.attr
```

For example:

```
root:::auths=solaris.*,solaris.grant; \\  
profiles=Web Console Management,All; \\ lock_after_retries=no; min_label=admin_low; \\  
clearance=admin_high
```



If you create a `user_attr` map in Active Directory, you must include the key as part of the value. For example:

## Importing and Managing NIS Maps

- Key: root
- Value: user name and properties followed by attributes defined as key and value pairs for the profile:  
all:::auths=solaris.\*;profiles=DBA,all;lock\_after\_retries=no

This map is only applicable for Solaris.

## Creating Generic Custom Maps

You can create generic maps to publish any type of custom information that you want to make available to NIS clients. Generic custom maps consist of a simple key/value format and optional comments. You can also use generic maps to manually create standar

To add a custom map to Active Directory:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, select **NIS Maps** and right-click; then click **New** and select **Generic Map**.
4. Type a name for the new map; then click **OK**.
5. In the details pane, select the new map, right-click; then click **New > Map entry**.
6. Type the appropriate information for the map record you are adding; then click **OK**. For example:
  - Type the **Key** to use in a client request for looking up the corresponding value.
  - Type the **Value** associated with the key.
  - Type any optional **Comments** for the key/value pair.

For example:



The screenshot shows a dialog box titled "New Map Entry". It has three text input fields: "Key:" with the value "salt.ajax.org", "Value:" with the value "127.0.0.1 salt.ajax.org localhost", and "Comments:" with the value "Sample hosts record". Below the "Value:" field, there is a small text box containing the instructions "Use \t to enter a tab character." and "Use \\ to enter a \ character.". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

## Changing the Map Type

When you import or create NIS maps, the map type determines the fields defined. For example, a Generic map type consists of three fields: the **Key** field (required) the **Value** field (required), and the **Comment** field. If you don't select the correct map type, the Centrify Network Information Service will not be able to interpret the records in the map correctly or respond to client requests with the proper information.

To change the map type of an existing NIS map:

## Importing and Managing NIS Maps

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, open **NIS Maps**; then select the map name you want to change. For example, if you have created a map named `nethosts`, select the `nethosts` map.
4. Right-click; then click **Change Type** and select the correct map type. For example, if the records in `nethosts` map should consist of a Key, a Value, and an optional Comment, select **Generic Map** as the map type.

If records have already been defined for the map using the incorrect map type, in most cases, you will need to modify the fields after changing the map type.

## Maintaining Map Records in Active Directory

Once NIS maps are stored in Active Directory, you must maintain the records in Active Directory to ensure changes are reflected in the local map cache that the Centrify Network Information Service uses to respond to NIS client queries. You can use Access Manager to manually add, edit, or delete individual map records for any map. The specific fields available in each record, and which fields are required and which are optional, depend on the type of map you are editing. For example, the fields in an `auto.master` map entry are different from the fields in a `netgroupmap` entry. For information about the fields in different types of maps, see *Creating new NIS maps in Active Directory*.

## Modifying Map Records in Active Directory

Specific users and groups can be given the right to add, modify, and delete NIS map entries using the Zone Delegation Wizard. For information about the rights required, see the *Planning and Deployment Guide*.

To edit individual map records:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, open **NIS Maps**, then select the map you want to modify. For example, select the `auto.master` map.
4. Select an individual map record and right-click.
5. Click **Properties** to modify the fields for the selected record or click **Delete** to remove the record from the map.

If deleting a map record, click **Yes** to confirm the operation.

## Deleting a map stored in Active Directory

Specific users and groups can be given the right to delete NIS maps using the Zone Delegation Wizard. For information about the rights required, see the *Planning and Deployment Guide*.

To remove a NIS map from Active Directory:

1. Open Access Manager.
2. In the console tree, select **Zones**, and open the specific zone you want to work with.
3. In the console tree, open **NIS Maps**, then select the map you want to remove.
4. Right-click; then click **Delete** to remove the map from Active Directory.

# Troubleshooting and Logging NIS Operations

This section describes how to use diagnostic tools and log files to retrieve information about `adnisd` operation and correct problems.

## Analyzing Zones for Potential Issues

One way to avoid problems with agentless authentication or incomplete information is to periodically analyze the zone in the Active Directory forest using the **Analyze** wizard.



When you run the **Analyze** wizard, it checks only *open* zones in the Active Directory forest. Make sure the zone you are using as a NIS domain is open before analyzing the forest.

To check for potential problems in the Active Directory forest:

1. Open Access Manager.
2. If so prompted, specify the forest domain or domain controller to which to connect.
3. In the console tree, select the Access Manager root node, right-click, and click **Analyze**.
4. At the Welcome page, click **Next**.
5. Select the checks to perform (at least the two in the table below) and click **Next**.

Select this option	To do this
Inconsistency in granting NIS server permissions	Check that a <code>zone_nis_servers</code> group exists in each zone that supports agentless authentication, and that the group contains all NIS servers defined for the zone (to ensure data integrity). This group is required for assigning permissions to Delinea-managed computers that act as NIS servers. Do not delete or modify it manually.
Orphan UNIX data objects	Check for profile objects whose parent objects have been deleted - for example, manually deleted zone objects whose user, group or computer UNIX profile data may be left in Active Directory. This option removes UNIX-specific data from Active Directory.

6. Review the summary report and click **Finish**.
7. If the summary report indicates any issues, select **Analysis Results** in the console tree and view the details listed in the right pane. For example:

To drill down further, or to resolve the issue, select the warning or error, right-click, and select **Properties**. For example:

## Verifying NIS Configuration for Servers and Clients

If you are troubleshooting issues with the Delinea Network Information Service or NIS client look-ups, start by verifying whether the current environment is configured properly by doing the following:

## Troubleshooting and Logging NIS Operations

- Check the connectivity between the NIS client and the NIS server with a `ping` command. If the `ping` command fails, check the network connection and the DNS configuration for name resolution problems.
- Verify that the `nisd.securenets` parameter allows responses to NIS clients on other computers. By default, the `adnisd` process responds only to `/local` NIS requests.
- Verify that the `adnisd` process is running, for example with the `ps` command. If `adnisd` is not running, restart it.
- Verify that `ypserv` is *not* currently running. If `ypserv` is running, stop it, modify the system initialization files so `ypserv` does not start when the computer is rebooted, and restart `adnisd`.
- Verify that `adnisd` has registered with RPC by running `rpcinfo -p localhost` on the `adnisd` server. You should see two entries in the RPC table for the `ypserv` program (100004):

program	vers	proto	port
100004	2	udp	844 ypserv
100004	2	tcp	846 ypserv

If no table is displayed, restart RPC services. If the `ypserv` process is not listed, restart `adnisd`.

- Verify RPC connectivity from the NIS client:

```
rpcinfo -p server
```

You should see the same table and entries as when you listed RPC entries for the `adnisd` server. For example:

program	vers	proto	port
100004	2	udp	844 ypserv
100004	2	tcp	846 ypserv

If no table is displayed, check the access permissions to the RPC server. For example, on Linux, check `/etc/hosts.allow` and `/etc/hosts.deny` files.

- Make sure the correct NIS domain name is configured on the NIS client. The NIS domain name is usually the same name as the name of the zone that the server is joined to. To set the domain name, log on as `root` run the following command:

```
domainname zone_name
```

- Verify that the `ypbind` process is running on the NIS client using the `ps` command. If `ypbind` is not listed as a running process, configure and start it.
- Verify that `ypbind` on the NIS client has found the Delinea NIS server by running `ypwhich` on the NIS client machine.

If the client is not bound to the correct server name, check the `ypbind` configuration files and start-up options.

If you are transitioning from an existing NIS infrastructure to the Delinea Network Information Service, the most common reasons for errors are an incorrect `domainname` setting or an improper `ypbind` configuration. For example, if your existing NIS domain names do not match the zone name, some clients may fail because they

use the old NIS domain name instead of the domain name you have set up for the Delinea Network Information Service domain.

### Updating the Startup Sequence

On some platforms, the `adnisd` package might prevent the `ypbind` service from starting properly because of the order in which services are started. For example, if `ypbind` is configured to start before the `adnisd` service, the `bind` will fail. In most cases, this issue does not occur if you are installing new packages because the installation process checks and corrects the startup sequence to ensure that the `bind` will be successful. However, to prevent unintended changes to the existing startup sequence during an upgrade, upgrading the `adnisd` package will not modify your existing startup configuration. You can manually correct the startup sequence after an upgrade by running the `chkconfig` script. For example, run the following command after the `adnisd` upgrade:

```
chkconfig adnisd on
```

### Using NIS Command Line Utilities

The Delinea Network Information Service supports common command-line utilities for performing administrative and diagnostic tasks. The following table lists those you may find useful in the Delinea NIS environment.

Use this command	To do this
<code>ypwhich</code>	Display the name of the NIS server the client is connected to.
<code>ypwhich -m</code>	List the maps that are served by the current NIS server.
<code>ypwhich -x</code>	Display the nicknames that are defined for NIS maps.
<code>ypcat -k map</code>	Display the contents of the specified map. This command displays both keys and values.
<code>ypmatch key map</code>	Look-up the specified key in the specified map.
<code>yppoll map</code>	Check the version number of the specified map. This command is only available on Solaris and HP-UX environments. The version number is displayed as an integer. The <code>adnisd</code> process does not use timestamps.

### Configuring Logging for `adnisd`

By default, the `adnisd` process logs errors, warnings, and informational messages in the `syslog` and `/var/log/messages` files, along with other kernel and program messages. You might find it useful to log additional details about the operation of the `adnisd` process for troubleshooting purposes.

To enable logging for the Delinea Network Information Service:



## Troubleshooting and Logging NIS Operations

1. As root, set the logging level for the Delinea Network Information Service by modifying the `log.adnisd` parameter in the `centrifdc.conf` file.

You might also want to suppress log messages from `adclient` to make it easier to collect and analyze the messages that are specific to `adnisd` operation. For example, set the `log.adnisd` parameter to `DEBUG` to log *all*`adnisd` operations, and the `log` parameter for `adclient` to `INFO` or `WARN` to limit messages generated by the `adclient` process:

```
log: WARNlog.adnisd: DEBUG
```

If you only want to collect diagnostic information for `netgroup` processing, set the `log.adnisd.netgroup` parameter instead of the `log.adnisd` parameter. For example:

```
log.adnisd.netgroup: DEBUG
```

2. Set the `syslog` facility to use for logging `adnisd` operations using the `logger.facility.adnisd` configuration parameter. This parameter enables you to log `adnisd` messages using a different `syslog` facility than the facilities used for logging general `adclient` messages or `adclient` audit messages.

This parameter value can be any valid `syslog` facility. For example, set this parameter to log messages to `auth` (default), `authpriv`, `daemon`, `security`, or `local0-7` facilities. For example:

```
logger.facility.adnisd: auth
```

For performance and security reasons, only enable `DEBUG` logging when necessary - for example, when requested to do so by Delinea Support, or while diagnosing a problem.



Sensitive information may be written to this file. Evaluate the contents before giving others access to it.