SecureLogin 8.7
Administration Guide

December, 2018

Legal Notice

Contents

About This Guide

This manual provides you information about administering SecureLogin. This manual contains the following sections.

Additional Documentation

For the latest version of SecureLogin guides, see www.netiq.com/documentation/securelogin/

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **Add Comment** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

1 Getting Started

NetIQ SecureLogin is an enterprise single sign-on product. It provides authentication solutions to Web, Windows, host, and legacy application-based single sign-on. NetIQ SecureLogin functions as an identity overseer for all the systems that users access.

It is a credential management tool developed to increase an organization's network security, while lowering support costs.

NetIQ SecureLogin securely manages and encrypts the authentication information in a directory. It stores usernames and passwords and automatically retrieves them for users, when required.

Recommendations

Before you begin configuring and administering NetIQ SecureLogin, it is recommended that you have a strong working knowledge of the following:

- Microsoft Active Directory
- Microsoft Management Console (MMC)
- Microsoft Group Policy Object Management Console (GPMC)
- Microsoft Windows operating systems
- Lightweight Directory Access Protocol (LDAP)

The Administrative Management Utilities

NetIQ SecureLogin consists of the following Administrative Management utilities and plug-in.

The utilities are:

iManager

iManager is a state-of-the-art Web-based administration console that provides customized secure access to network administration utilities and content from any location in the world. With a global view of your network from one browser-based tool, you can proactively assess and respond to changing network demands.

It is recommended to use iManager to administer NetIQ SecureLogin in Novell eDirectory environments.

IMPORTANT: Throughout this document, we refer to iManager as the Administrative Management Utility to explain the various administration procedures.

The graphics also represent an eDirectory configuration.

Starting iManager

Accessing iManager varies based on the iManager version (server-based or workstation) and the platform on which iManager is running. For more information on accessing iManager refer the iManager documentation (https://www.netiq.com/documentation/imanager/imanager_admin/data/bouqst6.html)

Accessing Server-Based iManager

1. Enter one of the following in the Address (URL) field of a supported Web browser:
   - **Default URL on non-OES 2 platforms:** `https://<server ip address>:8443/nps/iManager.html`

     On platforms other than Novell Open Enterprise Server 2 (OES 2), you must specify the Tomcat port as part of the iManager URL because iManager 2.7 uses only Tomcat 5 for its Web server requirements.
   - **Default URL for OES 2 platforms:** `https://<server ip address>/nps/iManager.html`

   Although slightly different, iManager URLs might work on some platforms. NetIQ recommends using these URLs for consistency.
2. Log in by using your username, password, and the tree name.

Accessing iManager on a Workstation

1. Browse to the iManager set up on your workstation.
2. Execute `imanager\bin\iManager.bat`.
3. Log in by using your username, password, and tree name.

SLManager

You can use SecureLogin Manager (SLManager) for any LDAP installations such as AD and eDirectory.

There is no difference in the features and components of iManager and SLManager. The menu options in both the utilities are similar. Only the user interfaces are different.

Starting SLManager

1 On the Start menu, select **Programs** > **NetIQ SecureLogin** > **SecureLogin Manager**. The Administrative Management utility is displayed.

NOTE: To use SLManager, Administrative Tools must be installed. Select the Administrative Tools option from the custom setup screen during the installation process. If Administrative Tools is not installed, use the Modify option from the installer to install it and start SLManager.

---

2 In the **Object** field, specify your object name, then press the Enter key.

You must press the Enter key to submit the entry typed in the Object field. Clicking OK closes the dialog box but does not accept the entry you typed. The object name should be in the LDAP convention (username, objectname) if you are using LDAP mode and in the eDirectory convention (username. objectname), if you are using the eDirectory mode.

Microsoft Management Console Snap-In

Use the Microsoft Management Console (MMC) snap-in for Active Directory deployments.

Starting MMC

1 On the Windows **Start** menu, select **Programs** > **Administrative Tools** > **Active Directory Users and Computers**. The Microsoft Management Console is displayed.

Known Issues

Note the following issue before you begin configuring and administering NetIQ SecureLogin.

Applications, Preferences, and Policies Added at the Group Level

The applications and policies added at the group level through iManager are not reflected on the client.

Every time a new group is created, you must re-assign the rights. You must manually assign read permissions for the correct functioning of the configured group.

Do the following on iManager for the applications, preferences, policies, and others added at the level to be reflected on the client:

1 Log in to iManager.

2 Select **Rights** > **Modify Trustees**.

3 Specify the object name.

4 Click **Add Trustee**. Browse and locate more objects.

Selection of multiple trustees is allowed.

5 Select **Assigned Rights** > **Add Properties**. Add the following attributes:

- Proto:SSO Entry
- Proto:SSO Entry Checksum
- Proto:SSO Security Prefs
- Proto:SSO Security Prefs Checksum

6 Click **OK**.

7 Click **Done** to save the changes and exit.

Migration from Existing Datastore to LDAP

Using the slMigrationHelper tool, if you attempt to modify the datastore from an existing one to LDAP, the datastore migration fails. This is because any LDAP or LDAPv3 mode requires NICI component to be installed.

Use the `-u` option to specify the path to SecureLogin installer. For example: `slmigrationhelper.exe -u C:\NetIQSecureLogin.exe -t LDAP -q`. This switches the datastore to LDAP and installs NICI in the quiet mode.

Additional Information

This section provides information about some of the SecureLogin functionality, and capabilities of the SecureLogin user interface.

Forcing Users to Change Password Before Grace Login Expires

SecureLogin allows administrators to force users to change their password before the grace login expires.

Scenario: SecureLogin is installed in LDAP mode with eDirectory. When the password expires, the authentication process consumes all the grace logins and users cannot log in. To avoid this, create the following registry keys.

- `GraceDaysBeforePasswordExpire` registry of DWORD value. This displays a warning message to the users about the number of days remaining for password expiry.
- `DaysForcePasswordChange` of DWORD value. This forces the users to change their password. Although the grace login available, this forces the users to change their password before the grace login expires.

For example, if the password policy is set to change every 90 days, the `GraceDaysBeforePasswordExpire` can be set to 5 and `DaysForcePasswordChange` can be set to 3. On the day 85 when users logs in, a message indicating the number of days left before password expiry appears. The users can choose to change the password immediately or change it later.

Similarly, when a users logs in on day 87 another message appears that forces the users to change the password. They cannot continue without changing the password.

NOTE: It is recommended to have the value of grace login to be more than 2. Since SecureLogin utilizes one grace login count for every connection with the directory, it is recommended to set the value of the grace login greater than 2.

Support for Oracle Forms

SecureLogin supports single sign-on to Oracle Forms that uses Java 1.7 or 1.8. If any of these Java components is added in the machine after installing (or upgrading to) SecureLogin, you need to enable SecureLogin to use the newly added Java component. To enable support to the new Java component, run the repair option of the SecureLogin installer.

Enhanced Application Definition Wizard and Theme Change

SecureLogin 8.0 SP1 and later includes more preferences that are added for specific sign-in options. The color of the theme is changed to blue and the color of the SecureLogin icon is also blue.

The SecureLogin Icon Changes Color Indicating Cached Application

NetIQ SecureLogin caches the applications that are launched, and are available for single sign-on. The user can click on the SecureLogin icon and view the list of opened applications. When the applications are cached, the color of the icon changes to orange. When you clear the list of applications, the icon changes to blue, which is the default color of the icon.

Notification Is Displayed

When you launch any application that is available for single sign-on, SecureLogin displays a notification in the system tray indicating that the application can be selected for single sign-on. If you do not want to single sign-on to an application when it is launched then, you can ignore the notification and proceed. When you ignore the notification, the color of the icon changes from blue to orange indicating that web page/ web pages are available for single sign-on. You can view the list of web pages by left clicking the icon once on the system tray and selecting the application to single sign-on. In Windows 10, instead of notification balloon a toast message is displayed.

Display SecureLogin User Name on the Task Bar

You can identify the active SecureLogin user with the help of the visual cue in the task bar. This visual cue displays the details of the active user such as First name, Last name, Full name, Distinguished name, or Default name based on the preference settings. To modify these preferences, refer “Display user name on task bar” on page 32. These preferences will be refreshed every 30 seconds by default. You can also modify the refresh time interval by modifying the value of the registry key `UserbarRefreshInterval`.

In addition, you can also add prefix to the user name displayed in the task bar. To add prefix, you must set the prefix text as a value for the registry key `UserbarPrefix`. These registry settings are available at `HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin`.

SecureLogin does not display the logged in user name by default in the task bar. For user name to be displayed in the task bar, you must right-click the SecureLogin icon on the notification area (system tray) and select `Show User bar` or you can rightclick on the task bar and select `Toolbars -> SecureLogin SSO User`.

Keyboard Shortcuts

You can now use the keyboard shortcuts to navigate to the required options.

To view the underlined letters in menu and dialog box options, press the Alt key, on the keyboard. Following is list of hot keys:

Key combination	Result
Ctrl+Shift+A	Launches the New Application window.
Ctrl+L	Launches the Create Login window.
Ctrl+P	Launches the New password policy window
Delete	Deletes the selected application, login, and password policy.

2 Configuring

User IDs, applications, and password policies must all have unique names and you cannot create an application named Error.

Before you deploy NetIQ SecureLogin, you must complete the following configuration tasks:

Setting User Preferences

You can set the user preferences in the Preferences page for any of the administrative management utilities. Each user preference has a default value that is implemented until an alternative value is set. In directory hierarchies, the preference values are inherited from higher level objects. However, the preference values set at the user object level override any values inherited from other objects in the directory.

You can restrict users from setting or modifying the preferences.

Changing a Preference Value

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Preference**. The Preference properties table is displayed.
3. Locate the setting you want to change and then, in the Value column, select the appropriate value.

   Some of the value settings are text field entries where you have to provide the value.
4. Click **OK**. The selected value is saved and the Administrative Management utility closes.

Disabling User Access

By default, the user has permission to change application definitions and predefined applications, passwords, and functionality. You do this through the administrative management utilities.

You have several options for restricting user access by setting preferences at the user, group policy, container, or organizational unit level. This includes:

- Full access to all administrative tools.
- Access to selected administrative tools.

- Hiding the SecureLogin icon on the notification area (system tray).
- Hiding and password protecting the SecureLogin icon in the notification area (system tray).

If the SecureLogin icon is password protected, anyone attempting to access the NetIQ SecureLogin Client Utility through the SecureLogin icon is prompted to provide the network password. This prevents non-authorized users from viewing SecureLogin data. However, an authorized user can use the administration tools to modify SecureLogin.

Updating the Datastore Objects

SecureLogin 8.5 supports the datastore versions 3.0, 3.5/ 5.5, and 6.0. The SecureLogin datastore version 6.0 includes a range of security features, including storing the single sign-on credentials on the user's smart card, encrypting the datastore by using the Public Key Infrastructure (PKI)-based credentials and the Advanced Encryption Standard (AES) encryption algorithm support.

To support the new features, you must change the SecureLogin datastore format to datastore version 6.0.

The SecureLogin datastore version 6.0, or later client can read data from the 3.x version of the datastore. However, the older versions of SecureLogin cannot read the data from the 6.0 and later versions of the datastore. In case of a mixed corporate environment in which some workstations are running SecureLogin 8.5, and the other workstations are running previous versions of SecureLogin, when a user moves between different versions of SecureLogin on different workstations the data gets incompatible. This is particularly a problem in Citrix environments or in large enterprise deployments.

If the SecureLogin datastore version 3.x is present when you install SecureLogin 8.5 then, SecureLogin detects 3.x data and continues to function correctly.

If you require the new functions, complete the following processes:

1. Choose a section of the tree to upgrade.

   For example:

   - Container
   - Group
   - Organization
   - User
2. Make sure that all user workstations in that section of the tree are upgraded with the latest SecureLogin client.

The next time the users log in, their data is converted to the latest version format and the new functions will be available.

Changing the Organizational Unit Level Datastore

- "Changing the Organizational Unit Level Datastore in an Active Directory Environment" on page 19
- "Changing the Organizational Unit Level Datastore in an eDirectory Environment" on page 19
- "Deploying an Upgrade" on page 19

Changing the Organizational Unit Level Datastore in an Active Directory Environment

Perform the following to set the directory datastore version at the organizational unit level in an Active Directory environment:

1. **On Microsoft Windows Vista:** On the Windows Start icon, select **All Programs** > **Control Panel** > **System Maintenance** > **Administrative Tools**.

   On Microsoft Windows XP: On the Windows **Start** menu, select **Programs** > **Administrative Tools** > **Active Directory Users and Computers**. The Microsoft Management Console is displayed.
2. Right click the required group policy, container, or OU, then click **Properties**. The properties dialog box is displayed.
3. Click the **SecureLogin** tab. The SecureLogin page is displayed.
4. Click **Manage**. The Advanced Settings page of the administrative management (SecureLogin Manager) utility is displayed
5. On the left pane, click **Advanced Settings**. The Advanced Settings page is displayed.
6. Click the **Datastore** tab.
7. From the **Select version drop-down** list, select the required version. A warning is displayed.

   The warning message refers to 3.0 clients that were used by some users who later upgraded the datastore mode to version 7.0.

   When a user's directory data version is upgraded, the datastore information displayed in the SecureLogin About box is not updated. To update this information, the user must activate Refresh Cache from the Advanced menu of the SecureLogin icon on the notification area (system tray).

Changing the Organizational Unit Level Datastore in an eDirectory Environment

1. Log in to iManager.
2. Click on **SecureLogin SSO** under roles and tasks.
3. Specify the organizational unit object.
4. Click **OK**.
5. Click **Advanced Settings**. The Advanced Settings page is displayed.
6. From the Select version under the **Datastore** section, select the required version. A warning is displayed.
7. Click **OK** to save the changes.

Deploying an Upgrade

When you are deploying an upgrade across a series of workstations, follow the procedure explained in "Changing the Directory Datastore" on page 20. The next time the directory server and the workstation caches are synchronized and SecureLogin operates in the new version mode.

Changing the Directory Datastore

When the directory is upgraded, the new features of SecureLogin are not available on the workstation. So, users must upgrade to the new version.

You can configure directory datastore version at the group policy, user object, container, or organizational unit levels. Set the datastore version at the container or the organization unit levels. This helps enterprises to manage the datastore base and minimize the possibility of conflicting versions.

If you require to update a single new feature of SecureLogin preference that requires 7.0 datastore, you are prompted with a warning before proceeding to change. For example, when you upgrade the **Use AES for SSO data encryption** preference.

Deleting or Re-setting User Data

If a user forgets the network password and passphrase answer (if the passphrase system is enabled), or if the login credential data has got corrupt, you can delete all the SecureLogin information for that user.

You as an administrator must do this because the user does not have access to the administrative management utilities.

Before you delete a user's datastore object, consider the following important aspects:

User Data Re-set Option	Action
Select the required directory object only	The Delete single sign-on configuration for this datastore object option is available at the container, group policy, ou, and user object level.
Record (external to SecureLogin) all usernames, password, and additional required credential information	For example, if you delete a single sign-on-enabled application at the ou level, you might also be deleting the credentials for all users that reside in that container.
Delete the local cache on the workstation	The object or user continues to inherit configuration from higher-level objects in the directory even though you deleted the user data in the directory cache. This means that you should delete the local cache on the workstation first. This ensures that it does not synchronize with the directory cache and re-create the configuration in the directory.

To reset the user data:

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. If you are using iManager, browse to **SecureLogin SSO** > **Manage SecureLogin SSO** > **Advanced Settings**. The Advanced Settings page is displayed.
3. Click **Delete** in the Datastore section. A warning message appears.
4. Click **Yes**. The Datastore object is deleted.

   If you did not delete the SecureLogin cache from the local cache, before you deleted the Datastore object data, you get an error message.

NOTE: When SecureLogin is installed in the SecretStore environment, the data will not be deleted from the SecretStore datastore.

In a SecretStore environment, use the SecretStore iManager plugin to clear the credentials instead of the SecureLogin plugin.

5 Click **Yes**.

When you do this, you delete the complete data of the user, including:

- Credentials, including usernames and passwords
- Application definitions
- Predefined applications
- Password policies
- Preferences
- Passphrase questions and answers

WARNING: The deleted data cannot be retrieved.

The next time the user logs in, the user is asked to set up a new passphrase question and response and re-enter the credentials for each application enabled for single sign-on.

NetIQ SecureLogin supports setting a cache expiry by using the following registry entry on the client:

`HKEY_LOCAL_MACHINE/SOFTWARE/Protocom/SecureLogin`

`DWORD` Value CacheExpiryDays

The value data is the number of days. Do not provide zero (o) because the cache would expire immediately on refresh. The cache expiry period is updated at each cache or directory synchronization, or each time SecureLogin loads in an online mode.

NOTE: No warning is provided at cache expiry. If a cache is expired, the users cannot access SecureLogin in an offline mode until they log in, and create the cache again in an online session.

3 Managing Preferences

NetIQ SecureLogin preferences are tools, options, and parameters used by the enterprise administrators to configure the user's SecureLogin corporate environment.

You can restrict a user's access to his or her SecureLogin preferences through the administrative management utilities.

The preferences also include applications that are permitted to be enabled for single sign-on and the tools to enable users to access their own SecureLogin management and administration functions.

You can configure user preferences from the Preference properties table in the administrative management utilities. This section provides information on the following:

Changes to Preferences

Rights for Installing JREs

SecureLogin now checks for new JREs installed on the client when SecureLogin is launched. If new JREs are detected and they are allowed by user's permissions, the new JREs are enabled for single-sign on automatically, without any user prompts or intervention.

The JRE update process requires local administrative rights on the client. If the user is not logged in with administrative rights, the update fails without displaying any notification to the user.

The JREs are updated the next time the user logs in with administrative rights and launches SecureLogin on the workstation.

Preferences Categories

The SecureLogin preferences are divided into the following categories:

- **.Net**
- **General**
- **Java**
- **Web**

- **Windows**
- **Wizard**

The Default Preference Values

Each preference value has a default value that is implemented during installation or deployment. You can configure alternative values.

Inheriting Preference Values

In corporate directory hierarchies, preference values are inherited from higher-level objects, while some lower-level objects can override preferences set at higher-levels.

Therefore, the preference values set at the user object-level override all higher-level object values.

Setting User Preferences

You can set the SecureLogin user preferences in the Preferences Properties table in the Administrative Management utilities (iManager, SecureLogin Manager, Microsoft Management Console).

Each SecureLogin preference has a default value that is implemented until an alternative value is manually configured. In directory hierarchies, preference values are inherited from a higher-level object, while some lower-level objects can override preference set at higher level.

For example, preference values set at the user object level override all higher level object values.

NOTE: This can be controlled for users by restricting their ability to set preferences.

For more information about inheriting configuration settings, see Chapter 13, “Distributing Configurations,” on page 97.

Changing Preference Value

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Preference**. The Preference properties table is displayed.
3. Locate the setting you want to change and then, in the **Value** column, select the appropriate value.

   NOTE: Some of the value settings are text field entries where you have to provide the value.

   ---

4. Click **OK**. The selected value is saved.

Setting the Preferences

You set preferences for managing SecureLogin in the Administration Management utility:

1. Log in to iManager.
2. Click **NetIQ SecureLogin** > **Manage SecureLogin** > **Preferences**. The list of preferences is displayed.
3. Make the changes you want, then click **OK**.

Use the information in the following tables to assist you in making the changes:

Table 3-1 *The .Net Preferences*

Preference	Possible Values	Description	Default Value
Allow single sign-on to WindowsAutomation (DotNet) applications	**Yes**/**No**/**Default**	This preference allows single sign-on to the .Net applications. If this preference is set to **No** then, the application will not be available for single sign-on.	The default value is **Yes**.
Add application prompts for WindowsAutomation (DotNet) applications	**Yes**/**No**/**Default**	This preference prompts to add a .Net application for defining the application definition. If this preference is set to **No** then, the Add Application window is not launched when you launch any .Net application.	The default value is **Yes**.
Start the WindowsAutomation (DotNet) monitor/ automation worker	**Yes**/**No**/**Default**	This preference will start the DotNetSSO process. The **Start the WindowsAutomation (DotNet) monitor/ automation worker** preference replaces the DISABLE_DOTNETSSO registry setting.	The default value is **Yes**.

NOTE: For the changes to be effective, restart SecureLogin after making changes to any of the **.NET** preferences

Table 3-2 *The General Preferences*

Preference	Possible Values	Description	Default Value
Allow "Close" option via system tray	**Yes**/**No**/**Default**	This preference controls whether users can access the **Close** option from SecureLogin icon on the notification area (system tray). If the option is set to **No**, the **Close** option is shown as disabled in the SecureLogin notification area (system tray) icon. If this option is set to Yes or **Default**, the **Close** option is displayed and accessible in the SecureLogin notification area (system tray) icon. **NOTE:** This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow "Refresh Cache" option via system tray	**Yes**/**No**/**Default**	This preference controls whether users can refresh cache using the **Advanced** > **Refresh Cache** option from the SecureLogin icon on the notification area (system tray). If this option is set to **Yes**, the **Refresh Cache** option is displayed and accessible in the notification area (system tray) icon. If this option is set to **No** or **Default**, the **Refresh Cache** option is not displayed in the notification area (system tray) icon. **NOTE:** This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **No**.

Preference	Possible Values	Description	Default Value
Allow "Log Off" option via system tray	**Yes**/**No**/**Default**	This preference controls if users can log out from a session using **Log Off User** option from the SecureLogin icon on the notification area (system tray). If this option is set to **No**, the **Log Off User** option is not displayed and accessible in the SecureLogin notification area (system tray) icon. If this option is set to **Yes** or **Default**, the **Log Off User** option is displayed and accessible in the SecureLogin notification area (system tray) icon. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow "Work Offline" option via system tray	**Yes**/**No**/**Default**	This preference controls whether users can work in offline cache mode using the **Advanced** > **Work Offline** option. If this option is set to **Yes** or **Default**, the **Work Offline** option is displayed in the notification area (system tray) icon. If this option is set to **No**, the **Work Offline** option is not displayed in the notification area (system tray) icon. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow application definition to be modified by users	**Yes**/**No**/**Default**	This preference controls whether users can modify application definitions using the **Definitions** tabs in the Applications pane of SecureLogin client. If this option is set to **Yes** or **Default**, the end user can view and modify their application definitions. If this option is set to **No**, the end user cannot change their application definitions. **NOTE:** If the Allow application definition to be viewed by users is set to No, then this option is cannot be edited. Disabling this preference does not disable the users from creating new applications through the wizards. This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default option is **Yes**.

Preference	Possible Values	Description	Default Value
Allow application definition to be viewed by users	**Yes**/**No**/**Default**	This preference controls whether users can view application definitions using the **Definitions** tabs in the Applications pane of SecureLogin client. If this option is set to **Yes** or **Default**, users can view the application definition. If this option is set to **No**, users cannot view the application definition. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow credentials to be deleted by users through the GUI	**Yes**/**No**/**Default**	This preference controls whether users can delete their credentials using SecureLogin client available from Manage Logins from the SecureLogin icon in the notification area (system tray). **NOTE:** If Allow credentials to be modified by users through the GUI is set to **No**, then this option is automatically set to **No** and not editable. This preference requires SecureLogin 6.0 datastore if the value is changed. If this option is set to **Yes** or **Default**, users can delete their credentials through the GUI. If this option is set to **No**, users cannot delete their credentials. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow credentials to be modified by users through the GUI	**Yes**/**No**/**Default**	This preference controls whether users can modify their credentials using SecureLogin client available from Manage Logins from the SecureLogin icon in the notification area (system tray). If this option is set to **Yes** or **Default**, users can modify their credentials through the GUI. If this option is set to **No**, users cannot modify their credentials through the GUI. They can only view the credentials. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Allow users to (de) activate SSO via system tray	**Yes**/**No**/**Default**	This preference controls whether users can activate or deactivate SecureLogin through the SecureLogin icon in the notification area (system tray). If this option is set to **Yes** or **Default**, users can switch between active and inactive modes of SecureLogin. If this option is set to **No**, users cannot switch between active and inactive modes. ◆ If SecureLogin status was active when this preference was applied, it remains as active and the user cannot de-activate SecureLogin. ◆ If SecureLogin status was inactive when this preference was applied, it remains as inactive and the user cannot change SecureLogin status to **Active**. This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow users to backup/restore	**Yes**/**No**/**Default**	This preference controls whether users can backup and restore their information from the **Advanced** menu of the SecureLogin icon on the notification area (system tray). If this option is set to **Yes** or **Default**, users can back up and restore their single sign-on information. If this option is set to **No**, users cannot back up and restore their single sign-on configuration. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Allow users to change passphrase	**Yes**/**No**/**Default**	This preference controls whether users can change their passphrase question and answer. The **Change Passphrase** option is available from the **Advanced** menu of the SecureLogin icon on the notification area (system tray). If this option is set to **Yes** or **Default**, users can change their passphrase through the notification area (system tray) icon. If this option is set to **No**, users cannot change their passphrase through the notification area (system tray) icon. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow users to modify names of Applications and Logins	**Yes**/**No**/**Default**	This preference controls whether users can edit the names of their Application login credentials using the **Details** tab > **Edit** function in SecureLogin client. If this option is set to **Yes** or **Default**, the user can edit the names of their credentials (either by right-clicking on the credential and selecting Rename, or by a slow double-click on the credential name). If this option is set to **No**, the use cannot edit the names of the credentials. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow user to set Obscure Show Password	**Yes/No/Default**	This preference controls whether the user can modify the **Obscure Show Password** option. If this option is set to **Yes** or **Default**, users can modify the **Obscure Show Password** option and change the duration of time a password is visible. If this option is set to **No**, users cannot modify the **Obscure Show Password** option.	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Allow users to view and change Preferences	**Yes**/**No**/**Default**	This preference controls whether users can view and update their preferences. If this option is set to **Yes** or **Default**, users can view and change their preferences. If this option is set to **No**, users cannot view and change their preferences. **NOTE:** Create a separate ou for administrators to ensure that they are not adversely affected by the general user configuration preferences at the ou level. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow users to view and modify API preferences	**Yes**/**No**/**Default**	This preference controls whether users can view and modify API options using the Preferences pane of SecureLogin client. The API preference defines the following options for users to: ◆ Enter an API license key(s). ◆ Provide API access. If this option is set to **Yes** or **Default** users can view and modify the API preference. If this option is set to **No**, users cannot view and modify the API preference. **NOTE:** This preference affects what is displayed in SecureLogin client using Change Preferences from the Advanced menu. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Allow users to view passwords	**Yes**/**Yes, per application**/**No**/**Default**	This preference controls whether users can view their passwords using Show Passwords in the Application pane > Details of NetIQ SecureLogin client. If this option is set to **Yes** or **Default**, users can view their passwords. If this option is set to **No**, users cannot view their passwords. **NOTE:** Allowing users to view their passwords gives them an opportunity to view and record passwords if they need to reset the SecureLogin configuration. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Change the cache refresh interval (in minutes)	**5**	This preference defines the time in minutes the synchronization of user data and directory on the local workstation. This preference is available in both SecureLogin client and the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is set to **5** minutes.
Display user name on task bar	**Disable**/**First name**/**Last name**/**Full name**/ **Distinguished name**/**Default**	This preference controls the display of the logged in user name on the task bar. If this option is set to **Disable**, the logged in user name is not displayed on the task bar. If this option is set to First name/Last name/Full name/Distinguished name/Default, based on the selection respective value is displayed on the task bar. **NOTE:** For the logged in user name to be displayed in the task bar, you must right-click the **Secure Login** icon on the notification area (system tray) and select **Show User bar** or you can right-click on the task bar and select **Toolbars -> SecureLogin User**. This preference is available in both the SecureLogin Client Utility and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Disable**.

Preference	Possible Values	Description	Default Value
Detect incorrect passwords	**Yes**/**No**/**Default**	Predefined applications generally include commands to respond to incorrect password dialogs. This preference enables SecureLogin to respond to incorrect passwords for web applications. If this option is set to **Yes** or **Default**, incorrect passwords for Web applications are detected. If this option is set to **No**, incorrect passwords for Web applications are not detected. This preference is available in both SecureLogin client and the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Disable single sign-on	**Yes**/**No**/**Default**	This preference controls the users access to running SecureLogin. If this option is set to **Yes**, access to SecureLogin is disabled and it will not start when run either automatically at startup or when run manually. If this option is set to **No** or **Default**, access to SecureLogin is enabled and will start normally. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **No**.
Display splash screen on startup	**Yes**/**No**/**Default**	This preference controls the display of the SecureLogin splash screen during startup. If this option is set to **Yes** or **Default**, the splash screen appears when SecureLogin starts up. If this option is set to **No**, the splash screen is hidden and users cannot see the splash screen when SecureLogin starts up. **NOTE:** This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Display the system tray icon	**Yes**/**No**/**Default**	This preference controls the display of SecureLogin icon in the notification area (system tray). If this option is set to **Yes** or **Default**, the SecureLogin icon appears on the notification area (system tray). If this option is set to **No**, the SecureLogin icon does not appear on the notification area (system tray). **NOTE:** When the SecureLogin icon is visible, users can double-click the icon on the notification area (system tray) to launch SecureLogin client. When the SecureLogin is not visible, users can start SecureLogin client through **Start** > **Programs** > **NetIQ SecureLogin** > **NetIQ SecureLogin** This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Display user name on mouse over tray icon	**Yes**/**No**/**Default**	This preference allows SecureLogin to display the current user name, when you mouse over the SecureLogin tray icon. When the user logs in to SecureLogin in the offline mode, the full qualified distinguished name (FQDN) is displayed when you mouse over the SecureLogin tray icon. In the online mode, the current user's full name is displayed.	The default value is **No**.
Enable cache file	**Yes**/**No**/**Default**	This preference controls creating and updating of a SecureLogin cache file on the local workstation. The cache file stores all user configuration data; local and inherited. Set this option to **Yes** or **Default**, the cache file is saved on the local workstation in the directory that was specified during install. Users with roaming profiles should always have this setting as **Yes**. Set this option to **No** if you cannot store cache files locally or if this causes conflicts with your organizational security policy. This preference is available in both SecureLogin client and the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Enable logging to Windows Event log	**Yes**/**No**/**Default**	This preference controls sending the log events to Windows Event Log. This includes the entire user configuration, both local and inherited. If set to **Yes** or **Default**, log events are sent automatically to Windows Event Log. If set to **No**, the log events are not sent to Windows Event Log. Only the following events are logged: ◆ SSO client started ◆ SSO client exited ◆ SSO client activated by user ◆ SSO client deactivated by user ◆ Password provided to an application by a script ◆ Password changed by the user in response to a change password command ◆ Password changed automatically in response to a change password command. **NOTE:** This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Enable the New Login Wizard on the system tray icon	**Yes**/**No**/**Default**	This preference controls whether users can create multiple logins on the same application using the **New Login** > **Add New Login** option from the NetIQ SecureLogin icon on the notification area (system tray). If this option is set to **Yes** or **Default**, the New Login menu option is enabled and users can create multiple logins. If this option is set to **No**, New Login menu option is disabled and users cannot create multiple logins. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Enforce passphrase use	**Yes**/**No**/**Default**	This preference forces users to set up a passphrase question and answer when SecureLogin is launched by a user for the first time. If this option is set to **Yes**, users must complete setting up their passphrase before they proceed with any other activity on the workstation. If this option is set to **No** or **Default**, users can postpone setting up the passphrase. If the users clicks **Cancel** or closes the dialog, then SecureLogin does not start. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **No**.
Enter API license key(s)	Specify API license key(s)	Specify the API license key(s) provided by SecureLogin to activate the API functionality for an application. You can add more than one API license key. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	Specify the API license key
Obscure Show Password (seconds)	Integer value in seconds	Restricts the password display time when you click the **Show Password** button in the local interface.	The default value is 30.
Password protect the system tray icon	**Yes**/**No**/**Default**	This preference restricts the users from accessing the NetIQ SecureLogin icon menu option (from the notification area (system tray) without their network login password. If this option is set to **Yes**, the NetIQ SecureLogin icon on the notification area (system tray) is password protected. If this option is set to **No** or **Default**, the NetIQ SecureLogin icon on the notification area (system tray) is not password protected. This preference is available in both SecureLogin client and the administrative management utilities (iManager, SLManager, and MMC snap-ins). **NOTE:** Always check the **Synchronize NDS password with Universal Password** option when NMAS is used.	The default value is **No**.

Preference	Possible Values	Description	Default Value
Provide API Access	**Yes**/**No**/**Default**	This preference controls the API functionality use. If this option is set to **Yes**, the API access is enabled. If this option is set to **No** or **Default**, the API access is disabled. This preference is available in both SecureLogin client and the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **No**.
Stop walking here	**Yes**/**No**/**Default**	This preference controls the inheritance of settings from higher level containers or organizational units. If this option is set to Yes, the inheritance of settings from higher level containers or organizational units is disabled. Set the option to **Yes** during phased upgrades when higher levels might have a different version of SecureLogin implemented. If this option is set to **No** or **Default**, the inheritance of settings from higher level containers or organizational units is enabled. This preference does not apply when SecureLogin is installed in eDirectory environment. The Corporate redirection functionality; that is, the inheritance settings from higher level container or organizational units is bypassed in an eDirectory environment. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **No**.
Wizard mode	**Administrator**/ **User**/**Disabled**	This preference controls that access to the application definition wizard. If this option is set to **Administrator**, it gives users' complete access to the application definition wizard. Users can create their own application definitions. If this option is set to **User**, users are only allowed to create new login credential sets for new applications using the auto-detection settings. If this option is set to **Disabled**, the application definition wizard is not launched. **NOTE:** This preference requires SecureLogin 6.0 datastore if the value is changed. This preference is available through the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Administrator**.

Table 3-3 *The Java Preferences*

Preference	Possible Values	Description	Default Value
Add application prompts for Java applications	**Yes**/**No**/**Default**	This preference prompts to add a Java application for defining the application definition. If this preference is set to **No** then, the Add Application window is not launched when you launch any Java application.	The default value is **Yes**.
Allow single sign-on to Java applications	**Yes**/**No**/**Default**	This preference controls whether SecureLogin allows single sign-on for Java applications. If the preference is set to **Yes** or **Default**, SecureLogin prompts the user to enter credentials (if none already exist), or submits existing credentials on the Java application login page. If this option is set to **No**, Java applications are not enabled for single sign-on. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Table 3-4 *The Web Preferences Properties Table*

Preference	Possible Values	Description	Default Value
Add application prompts for Internet Explorer	**Yes**/**No**/**Default**	This preference controls the display of the Web login detection wizard and confirmation dialog box when a Web application is detected and recognized by Internet Explorer. If you select **Yes** or **Default**, the user is initially prompted to enable the application and enter the credentials for the application (if not done previously). **NOTE:** Setting the preference to **Yes** when displayed to users depends on the settings of the Wizard mode preference. On subsequent runs of the application, the user is not prompted for credentials and single sign-on occurs seamlessly. If you select **No**, SecureLogin skips enabling the application for single sign-on, the user is never be prompted to enable the application. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Add application prompts for Mozilla Firefox	**Yes**/**No**/**Default**	This preference controls the display of Web login detection wizard and confirmation dialog box when a Web application is detected and recognized by Mozilla Firefox. **NOTE:** Setting the preference to **Yes** when displayed to users depends on the settings of the Wizard mode preference. If you select **Yes** or **Default**, the user is initially prompted to enable the application and enter the credentials for the application (if not done previously). On subsequent runs of the application, the user is not prompted for credentials and single sign-on occurs seamlessly. If you select **No**, SecureLogin skips enabling the application for single sign-on for this instance. You are prompted to enable the application when you launch it the next time. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.

Preference	Possible Values	Description	Default Value
Add application prompts for Google Chrome	**Yes**/**No**/**Default**	This preference controls the display of Web login detection wizard and confirmation dialog box when a Web application is detected and recognized by Google Chrome. **NOTE:** Setting the preference to **Yes** when displayed to users depends on the settings of the Wizard mode preference. If you select **Yes** or **Default**, the user is initially prompted to enable the application and enter the credentials for the application (if not done previously). On subsequent runs of the application, the user is not prompted for credentials and single sign-on occurs seamlessly. If you select **No**, SecureLogin skips enabling the application for single sign-on on this instance. You are prompted to enable the application when you launch it the next time. This preference is available in both the SecureLogin Client Utility and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Add application prompts for web pages on mutation	**Yes**/**No**/**Default**	This preference controls the display of Web login detection wizard and confirmation dialog box when a Web application includes mutation events. **NOTE:** Setting the preference to **Yes** when displayed to users depends on the settings of the Wizard mode preference. If you select **Yes** or **Default**, the user is not prompted to enter the credentials on the subsequent web pages that include mutation events but initially is prompted to enable the application and enter the credentials for the application (if not done previously). If you select **No**, SecureLogin skips enabling the application for single sign-on for the web pages that include mutation events. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **No**.

Preference	Possible Values	Description	Default Value
Allow single sign-on to Internet Explorer	**Yes**/**No**/**Default**	This preference defines single sign-on access to Web application using Internet Explorer. If you select **Yes** or **Default** the specified credentials are saved and the application is enabled for single sign-on. If you select **No**, SecureLogin does not prompt for credentials (if none exist or are incorrect) and does not submit credentials into the application. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow single sign-on to Mozilla Firefox	**Yes**/**No**/**Default**	This preference defines single sign-on access to Web application using Mozilla Firefox. If you select **Yes** or **Default** the specified credentials are saved and the application is enabled for single sign-on. If you select **No**, SecureLogin does not prompt for credentials (if none exist or are incorrect) and does not submit credentials into the application. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow single sign-on to Google Chrome	**Yes**/**No**/**Default**	This preference defines single sign-on access to Web application using Google Chrome. If you select **Yes** or **Default** the specified credentials are saved and the application is enabled for single sign-on. If you select **No**, SecureLogin does not prompt for credentials (if none exist or are incorrect) and does not submit credentials into the application. This preference is available in both the SecureLogin Client Utility and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Enable DHTML monitor on web pages	**Yes**/**No**/**Default**	This preference enables single sign-on for Web pages that require DHTML script. You can enable or disable the DHTML setting. For DHTML applications that depend on mutation events, the **Add application prompts for web pages on mutation** preference must be set to **Yes** to single sign-on to the application.	The default value is **Yes**.
Start the Flash monitor/ automation worker	**Yes**/**No**/**Default**	This preference starts the flash single sign-on process.	The default value is **No**.

Table 3-5 *The Windows Preferences Properties Table*

Preference	Possible Values	Description	Default Value
Add application prompts for Windows applications	**Yes**/**No**/**Default**	This preference controls the display of a Windows login detection and confirmation message when a Windows application is detected and recognized. If you select **Yes** or **Default**, the user prompted to enable the application and to enter the credentials for the application (if not done previously). On subsequent runs of the application, the user is not prompted for credentials and single sign-on occurs seamlessly. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Allow single sign-on to Windows applications	**Yes**/**No**/**Default**	This preference defines single sign-on access to Windows applications. If you select **Yes** or **Default** the specified credentials are saved and the application is enabled for single sign-on. If you select No, SecureLogin will not prompt for credentials (if none exist or are incorrect) and will not submit credentials into the application. This preference is available in both SecureLogin client and all the administrative management utilities (iManager, SLManager, and MMC snap-ins).	The default value is **Yes**.
Start the Windows 32bit (WinSSO32) monitor/ automation worker	**Yes**/**No**/**Default**	This preference starts the WinSSO32 process.	The default value is **Yes**.
Start the Windows 64bit (WinSSO64) monitor/ automation worker	**Yes**/**No**/**Default**	This preference starts the WinSSO64 process for Windows on a 64-bit machine.	The default value is **Yes**.

NOTE: For the changes to be effective, restart SecureLogin after making changes to any of the **Windows** preferences

Table 3-6 *The Wizard preferences*

Show Add Application wizard with minimal actions	**Yes**/**No**/**Default**	This preference controls displaying the prompt for single sign-on with minimum options. If you set this preference to **Yes**, the Add Application wizard displays the following minimal options: ◆ `Yes, I want to single sign using the default selections done by the wizard.` ◆ `Cancel, I do not want to single sign this screen at this time.` ◆ `No, never prompt me to single sign this screen.` If you set this preference to **No** then, the default options are displayed to single sign-on to the application.	The default value is **No.**
Skip the wizard process and use defaults for new forms	**Yes**/**No**/**Default**	This preference controls the wizard process. If you set this preference to **Yes**, then the default wizard selections are applied for all the pages of the application and you can switch between the panes instead of making changes sequentially.	The default value is **No.**

Configuring the Exclude.ini File

An admin can configure the `exclude.ini` file to exclude applications or classes of applications that do not require single sign-on. For example, exclude an anti-virus application because it does not require single sign-on. The admin can configure the `exclude.ini` file to achieve the following outcomes:

1. Exclude heavy applications that might cause performance issues.
2. Exclude the classes that do not require single sign-on to improve the performance of applications.

IMPORTANT: SecureLogin does not process the applications or classes specified in the `exclude.ini` file for single sign-on even if application definition is published for these applications.

By default, for optimal performance, the following executables are not processed for single sign-on by SecureLogin:

```
msdev.exe
slbroker.exe
tlaunch.exe
slproto.exe
notes.exe
nswebsso.exe
nwadmn32.exe
nwadmnnt.exe
nwadmn95.exe
loginw95.exe
setup.exe
nwtray.exe
loginw32.exe
scrnlock.scr
wfica32.exe
mmc.exe
slwinsso.exe
slmanager.exe
sllock.scr
```

Perform the following steps to configure the `exclude.ini` file:

1. Create an `exclude.ini` file in the `C:\Program Files\NetIQ\SecureLogin` directory.

   IMPORTANT: If you are using Notepad to create or edit the exclude.ini file, make sure to set the encoding to Unicode.

2. Specify the application executables or classes in the `exclude.ini` file in the following format:

   ```
   Include
   executable1.exe
   executable2.exe

   Classes
   class1
   class2

   Nodefault
   Exclude
   default1.exe
   default2.exe
   ```

In this example configuration of `exclude.ini`, SecureLogin will perform the following tasks:

1. SecureLogin will not process `executable1.exe` and `executable2.exe` for single sign-on.
2. SecureLogin will not process `class1` and `class2` for single sign-on.
3. SecureLogin will process `default1.exe` and `default2.exe` for single sign-on. It will override the default configuration.

4 Managing Passphrases

This section provides information on the following:

About Passphrases

Passphrases are an important security component in the implementation of SecureLogin. Passphrases are unique question and answer combinations created to verify and authenticate the identity of a user. In a directory environment, you can create passphrase questions for users. Users can select one of these questions and provide an answer for it. You can also permit users to provide a question of their choice and the answer for it.

Passphrases protect user credentials from unauthorized use. For example, in an Microsoft Active Directory environment, an administrator can reset the users network password and then log in as that user and gain access to the users information.

However, this cannot happen when you are using SecureLogin. If someone other than the actual users tries to reset the network password, SecureLogin triggers the passphrase question. The user must provide the correct answer before successfully logging in. Even an administrator cannot access the user’s single sign-on-enabled applications without knowing the user’s passphrase answer.

When SecureLogin is launched for the first time on a user’s workstation, the Passphrase Setup dialog box is displayed.

NOTE

- In a Microsoft Windows Vista or higher environment, when you log in to SecureLogin in an offline mode with an incorrect password, you are prompted to provide the passphrase answer. If an incorrect passphrase answer is specified, you are prompted to retry the authentication. However, if you again provide a wrong password, instead of seeing a prompt for the passphrase answer, you are prompted to specify the password (that is, instead of the passphrase dialog box, the password dialog box is displayed). Close and relaunch SecureLogin to be prompted for the password first, then prompted for the passphrase answer if the incorrect password is specified
- SecureLogin using the Novell Client does not support non-password-based NMAS logins if the passphrase options are disabled. This is not supported because SecureLogin either fails to open the local cache or opens the local cache file without any password.
- Also, Offline authentication does not work if you do a non-password-based NMAS authentication with the Passphrase Security System disabled. This is because SecureLogin in offline mode accepts only passphrases for non-password-based NMAS authentication. This scenario occurs only if SecureLogin is installed in Novell Client mode.

Passphrase Authentication

Passphrases are used to authenticate when:

- A user is working either remotely or offline in an eDirectory or non-Microsoft Active Directory LDAP environment.
- Someone other than the actual user resets the network password.

Benefits of Passphrases

Some of the benefits of using passphrase include:

- An individual cannot access a user's credentials by resetting the network password.
- Passphrases can be used in conjunction with SecureLogin Self-Service Password Reset, which enables users to reset their network password after answering the passphrase question.
- You can use this functionality to disable access to user credentials if the computer is stolen.

NOTE: You can disable the passphrase security system, but it also removes the features mentioned in the preceding section.

Creating a Passphrase Question

As an administrator, you can:

- Create one or more passphrase questions for users to select.
- Enable users to create their own passphrase question and answer.
- Set up a combination of both.

To create a passphrase question:

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Advanced Setting**. The advanced setting options are deployed.

   By default, **User-defined passphrase questions** is selected. Deselect this option if you do not want users to create their own passphrase question and answer.
3. Click **New**.
4. In the Enter a new passphrase question dialog box, provide your passphrase question.
5. Click **OK**. The question you provided is displayed in the **Corporate passphrase questions** field.

   This passphrase question is displayed to all users associated with the selected object.
6. Repeat the Steps 3 to Step 5 to create additional passphrases.

IMPORTANT: Make sure you click **OK** after you have created the passphrase question to save the changes and exit the page.

The passphrase answer is specified by the user when he or she sets up the passphrase question and answer. Ideally, passphrase answers must contain a minimum of six characters. However, you can change the policy to suit your security requirement. For more information, see "Changing a Passphrase Policy" on page 49.

We recommend that you do not apply strict policies to passphrase answers as it make them harder to remember. Instead, we recommend you to use a multivalue question, such as `What is you driving license number plus Your Mother's name?` and set a passphrase policy based on that.

Re-setting a Passphrase Answer

If a user forgets the passphrase answer, you must reset the user's SecureLogin configuration to ensure that the user's data is secure. This deletes all user-specific information, including usernames and passwords.

For more information on re-setting user data, see "Deleting or Re-setting User Data" on page 20.

IMPORTANT: When you set up a user's passphrase question and answer policies, we recommend that you keep them simple so that the user can easily remember the answer.

Changing the Passphrase Prompt

You can change the passphrase prompt that users see in the Passphrase Setup dialog box the first time they log in.

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Advanced Settings**. The Advanced Settings options are displayed.
3. Under **Customized Passphrase Prompt**, select the **Modify the passphrase prompt window text** check box. The **Custom prompt** is now active.
4. Specify the new prompt.
5. Click **OK** to save the changes and close the Administrative Management utility. Log in as a new user to view the customized prompt.

Changing a Passphrase

Users can change their passphrase answer depending on how you configure SecureLogin.

1. Right-click the SecureLogin icon in the notification area (system tray), then select **Advanced > Change Passphrase**. The Passphrase dialog box is displayed.
2. Specify the passphrase answer in the field.
3. Click **OK**. The Passphrase Setup dialog box is displayed.
4. In the **Enter a question** field, select or specify a passphrase question.
5. In the **Enter the answer** field, specify the new passphrase answer.
6. In the **Confirm the answer** field, retype the new passphrase answer.
7. Click **OK**.

NOTE: Users who do not have access to the SecureLogin icon cannot change their passphrases. You can temporarily enable access to the icon to allow the user to change the passphrase with the Display the system tray icon preference setting.

5 Managing Passphrase Policies

About Passphrase Policies

A passphrase is an integral part of the security architecture of SecureLogin. It can be used to secure single sign-on data when a user authenticates to applications.

A policy is used to restrict the format and content of passphrase answers, including length, whether numeric characters are required, and whether passphrases must be uppercase or lowercase. For example, you could set **Begin with an uppercase character** to **Yes** and **Maximum uppercase characters** to **1**, and **Prohibit characters** to **Disallow spaces**. You could also require all passphrase answers to start with uppercase and have the rest of the characters as lowercase.

You can set passphrase policies using any of the supported administrative management tools (iManager,MMS or slManager).

Changing a Passphrase Policy

The passphrase policy now applies to all users inheriting configuration from the selected object. You can change or disable it at any time.

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Advanced Settings**. The Advanced Settings options are displayed.
3. Select the **Use a passphrase policy** check box.
4. Click **Edit Policy.** The Passphrase Policy settings page is displayed.
5. In the **Setting Description** column, click the policy rule you want to edit, then in the **Value** column, specify the required value.

   For example, if you think that users might find it easier to remember basic rules for all passphrases instead of remembering exactly how they typed a passphrase when they created it, you could require all passphrases to contain a minimum of 6 characters and a maximum of 12 characters. Set **Minimum length** to 6 and set **Maximum length** to 12.

   By default, passphrase responses are required to contain a minimum of six characters. For security reasons, any passphrase policy you implement must also contain a minimum of six characters.
6. When you have finished setting the values in the table, click **OK**. The new values are added to the **Value** column.

Enabling the Passphrase Security System

This section contains information on the following:

The **Enable Passphrase Security System** option determines if users can use a passphrase to decrypt single sign-on data.

If the passphrase system is not used, this exposes the users’ single sign-on data if a third party can to reset the users network password. It is strongly recommended you enforce passphrase system on users environment.

To view or modify this preference:

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Preferences**. The Preferences page is displayed.
3. Select **Security** > **Enable passphrase security system** and from the drop-down list, select either **Yes** or **Hidden**.
4. Click **Apply**.
5. Click **OK**.

You can set the **Enable Passphrase Security System preference** to **Yes** or **Hidden** depending on the enterprise security requirements.

If the **Enable Passphrase Security System** is set to **Yes,** (which is the default preference) the user is prompted to set the passphrase question and answer when SecureLogin is launched for the first time.

If the **Enable Passphrase Security System** is set to **Hidden**, the user is not prompted to set the passphrase question and answer when SecureLogin is launched for the first time.

WARNING: If you change the preference from **Hidden** to **Yes**, users must answer the passphrase questions to use SecureLogin. Typically, users not prompted to create a passphrase after the first login.

Without any message indicating the change in the preference, users are prompted for the passphrase answer. So, avoid changing the preference.

You have two options, depending on what you specified.

- Users can create both the passphrase question and answer.
- You predefine a list of questions and answers, and the user selects from the list.

When users have set a passphrase, the application generates a random key, and a one-way hash of the passphrase answer encrypts this key. Later, the application key encrypts the new key. This key protects users’ SecureLogin credentials and passwords so that even someone with Supervisor rights to the network and access to Microsoft Management Console (MMC) is unable to view a user's passwords to applications.

After the passphrase is set, every time a user logs in to the network, SecureLogin loads seamlessly.

Typically, the prompt to create a passphrase is never seen after the first login. However, if an administrator resets the user's directory or network, the next time SecureLogin launches, users must answer the passphrase question before SecureLogin continues. This prevents other users from changing the user's directory password, logging on as the user, obtaining access to the SecureLogin data, and using it to run applications.

Enabling the Passphrase Security Using PKI Encryption

You cannot toggle the **Enable Passphrase Security System** setting when the users forget their smart card unless they had previously set a passphrase or had it randomly generated using the **Hidden** option.

If users are required to authenticate to the network by using passwords, **Enable Passphrase Security System** must be set either to **Yes** or **Hidden**.

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Preferences**. The Preferences page is displayed.
3. Under **Security**, select either **Yes** or **Hidden** in the **Enable passphrase security passphrase** drop-down list.
4. Click **Apply**.
5. Click **OK**.

If you select **Yes**, users must select a passphrase question and answer when they log in to SecureLogin for the first time. When the passphrase system is enabled, users are prompted to answer their passphrase question if their password has been reset by the administrator.

NOTE: With the **Use smart card to encrypt SSO data** option selected (either **PKI credentials** or **Key generated on smart card**), you can use the passphrase to decrypt single sign-on data if the user's smart card is damaged or lost.

This setting must be used in conjunction with the **Lost card scenario** preference set to **Allow passphrase** and **Store credentials on the smart card** preference set to **No**. You can toggle these preferences if the user's smart card is forgotten providing the user's passphrase has already been set. The user is prompted to answer the passphrase question before SecureLogin loads.

For more information, see "Lost Card Scenarios" on page 74.

If the **Hidden** preference is selected, users are not prompted to set a user-defined passphrase. A user key is generated automatically with any input from the user.

The **Enable Passphrase Security System** cannot be set to **No** unless **Use smart card to encrypt SSO data** is set to **PKI credentials.**

If users are required to authenticate to the network by using passwords, the **Enable passphrase security system** option must be set to **Yes** or **No** or **Hidden**.

IMPORTANT: With the passphrase security system set to **Hidden**, a directory administrator can reset a user's directory password, log in as the user, and access the user's single sign-on data because they are not prompted to answer a passphrase question.

Enabling the Passphrase Security Using PKI Encryption

If the **Use smart card to encrypt SSO data** is set to **PKI credentials**, the user's single sign-on data is encrypted by using the public key from the selected certificate and the private key and stored on a PIN-protected container on the user's smart card. Both, the user's directory datastore and the local cache are now protected by the PKI credentials.

The single sign-on data can be encrypted by using the private key that is PIN-protected and stored on the user's smart card for added security. Only the user who has the physical possession of the smart card and knowledge of the PIN can decrypt the single sign-on data.

To set the **Use smart card to encrypt SSO data** preference:

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Preferences**. The Preferences page is displayed.
3. Select **Security** > **Use smart card to encrypt SSO data** and from the drop-down list, select either **PKI credentials** or **Key Generated On Smart Card** or **No**.
4. Click **Apply**.
5. Click **OK**.

If the **Use smart card to encrypt SSO data** is set to **PKI credentials**, the **Enable passphrase security system** can be optionally set to **No**.

If the **Use smart card to encrypt SSO data** is set to **No**, the user's passphrases are completely disabled and the user's smart card is always required to decrypt the single sign-on data.

IMPORTANT: If your enterprise chooses to disable the passphrase security system:

- You can still access a user's credentials by resetting the network password.
- The functions of using the passphrases in conjunction with SecureLogin Self Service Password Reset (SLSSPR) is disabled. The SecureLogin Self Service Password Reset enables a user to reset his or her network passwords after answering the passphrase questions.

The supported directory modes for disabling the passphrase security system are:

- Active Directory
- LDAP-compatible
- eDirectory (if SecretStore is used)

For detailed information on the likely scenarios that a user might experience in environments where the **Enable passphrase security system** option is set to **No**, see "Passphrase Security System Scenarios" on page 53.

Checking the Passphrase Security System Status

1. On the notification area (system tray), right-click the NetIQ SecureLogin icon > **About**. The About dialog box is displayed.

   The status appears next to the `Database Mode` and is listed as either `PP Enabled` or `PP Disabled`.

Passphrase Security System Scenarios

The information provided in this section describes the user experience in environments where the passphrase security system has been enabled and disabled.

Scenario 1: The passphrase security system is disabled in a previously enabled environment

When the passphrase security system is disabled in an environment where it was previously enabled, the following message appears to users the next time they log in.

If the user clicks **OK**, the disabling of the passphrase security system is approved and the user is prompted for the current password. The approval is complete when the user provides the password.

If the user click **Cancel**, the passphrase security system disabling is delayed and the user is prompted with the message until he or she clicks **OK** to approve the change.

NOTE: Users must answer the passphrase answer to prevent the administrators to toggle this preference and allow an unauthorized user access SecureLogin.

Scenario 2: The passphrase security system is re-enabled in a previously disabled environment

If the passphrase security system is re-enabled, the Passphrase Setup dialog box is displayed (similar to when a user logs in for the first time after installing SecureLogin.)

If the user clicks **OK**, the user resets the passphrase question and answer.

If the user clicks **Cancel**, there is a delay in enabling the passphrases for the user's workstation. The user is prompted at subsequent log ins until he or she specifies a passphrase question and answer.

Scenario 3: The passphrase security system is disabled and the user has changed his or her passwords (restrictions for moving user objects)

If you reset the user's password when the passphrase security system is disabled:

- In an LDAP-compatible and eDirectory (with SecretStore) modes, you cannot move the user object to another organizational unit until that user has logged in to SecureLogin on his or her workstation. You must move the object back to its previous location to enable the user to run SecureLogin.
- In an Active Directory mode, you can move the user object within the directory. If the user object is moved, you must move the object back to its previous location to enable the user to run SecureLogin.

Scenario 4: Forgotten Passphrase

If a user forgets SecureLogin data, including his or her passphrase or passphrase answer, you must delete the user's existing SecureLogin datastore.

After the datastore is deleted, the user's corporate applications, credentials, preferences, and user policies are permanently removed. You must then reset the user's corporate password before he or she can log in and reconfigure the applications by using SecureLogin.

The next time SecureLogin starts, he or she must manually log in. SecureLogin then detects that a passphrase is not set and prompts the user to set up the passphrase before continuing. You can create a list of predefined list of passphrases questions.

After the user has set a new passphrase, he or she must re-enter the application usernames and passwords. If it is not done, an unauthorized could breach security by clearing the passphrase, entering a new passphrase, and accessing the actual user’s credentials.

You might need to reset the user’s application passwords as they might have forgotten them.

6 Managing Credentials

This section provides information on the following:

About Credentials

The first time a user logs in after creating an application definition and activating it for single sign-on, the user is prompted to provide credentials in a SecureLogin dialog box. SecureLogin then stores and associates these credentials with the application definition and uses it in subsequent logins.

Because individual application requirements determine the credentials that users must enter when manually logging in, only those credentials are stored and remembered by SecureLogin. For example, if users have an application that only requires username and password, SecureLogin encrypts and stores the username and password for subsequent logins. Alternatively, some applications require the user to enter domain and database names, IP addresses, and select various options on Web pages. SecureLogin can handle all these on behalf of the user.

You can display and manage these credentials in the **Logins** page of the Administrative Management utility and the **My Logins** pane of the SecureLogin Client Utility.

Credentials stored in a directory environment apply to all associated objects. For example, if users access an application located on a specific domain, and they are required to manually select or provide the domain address, then the domain must be configured as a credential in the **Logins** pane at the organizational unit level. Thereby, users need not manually provide the domain location when they log in. You can then change the domain at any time without notifying users.

Application credentials such as e-mail, finance system, human resource system, and travel system are typically stored for user objects and apply only to (and can be used by) the particular user. For example, John’s application credentials are encrypted and stored against John’s user object and only available to him. When he starts an application, SecureLogin retrieves, decrypts, and enters the credentials on behalf of John.

Creating a User Login and Credentials

Logins and credentials are typically created automatically as part of the application definition, but you can manually create and edit them, if required.

To create logins and credentials:

1. On the notification area (system tray), double-click the SecureLogin icon. The SecureLogin Client Utility is displayed.
2. Click **My Logins**. The existing Logins are displayed.

3 Click **New**. The Create Login dialog box is displayed.

4 In the **Name/Id** field, specify a Name/ID for the login.

5 Click **OK**. The Login name/ID is added to the My Logins pane.

6 From the My Logins on the left pane, select the login you have created.

7 Click **New**. The Create Credential dialog box is displayed.

8 In the **Name** field, specify a name for the new credential.

9 Click **OK.** The new credential is added to the Login details.

10 In the Value column, specify a value for the credential.

11 Click **Apply**. The new credential variable and its value are displayed.

Deleting a User Login and Credentials

An admin can delete the single sign-on configuration for a user using the SecureLogin Manager (SLManager). This includes deleting the applications, logins, and credentials of a user. Perform the following task to delete the single sign-on configuration:

1 Launch SLManager.

2 Click the username for which you want to delete the single sign-on configuration.

3 Click **Advanced Settings**.

4 Click **Datastore**.

5 In the **Single sign-on configuration** pane, click **Delete**.

IMPORTANT: If SecretStore is configured with SecureLogin, you must delete the logins and credentials of the user from the SecretStore vault. For more information on how to delete the logins and credentials from the SecretStore, see Removing a Secret in the SecretStore Administration Guide (https://www.netiq.com/documentation/secretstore34/nssadm/data/admu1ef.html).

Linking a Login to an Application

You can link a login to an application in the appropriate Login pane. For example, if users are logging in to Microsoft Outlook using a set of credentials and they are also logging in to Outlook Web Access, then they can share or link the credentials to the Web login application definition.

To link a login to an application:

1 In the notification area (system tray), double-click the SecureLogin icon on the system tray. The SecureLogin Client Utility is displayed.

2 Click **My Login** and the login that you want to an application.

3 Click **Link** icon. The Applications List dialog box displays the list of enabled predefined applications and application definitions.

4 Select the application that you want to link.

5 Click **OK**. The linked application is added.

6 Click **OK** to save changes and close the SecureLogin Client Utility.

Recovering a Forgotten Password

The Client Login Extension 3.7 provides password recovery support for applications that are accessed through SecureLogin. The password recovery support is available for graphical authentication interfaces such as GINA/Credential Provider for LDAP clients, Novell Client, and Microsoft clients. Clients in the Windows 7 and Windows Vista operating systems support Credential Provider model of graphical authentication interface. Clients in other operating systems support GINA/Credential Provider model of graphical authentication interface.

NOTE: To enable the Client Login Extension support for Novell Client in a SecureLogin setup, install Novell Client before installing the Client Login Extension tool.

The password recovery support through Client Login Extension tool is also available for locked workstations and for workstations in which user operations are controlled by Desktop Automation Services (DAS).

To know more about using Client Login Extension tool for recovering forgotten passwords, see the Client Login Extension Guide (https://www.netiq.com/documentation/idm401/idm_cle/data/bg4sw5i.html).

7 Managing Password Policies

SecureLogin provides the password policy functionality to enable you to efficiently and effectively manage user passwords, in order to comply with your organization's security policies.

This section provides information on the following:

About Password Policies

You can create password policies at a container, OU, Group Policy, and user object level. Policies set at the container or organizational unit level are inherited by all associated directory objects. Password policies set at the user object level override all higher-level policies. Password policies are linked to application definitions through scripting and are not applied to directory objects. You can do this by creating a password policy in the Password Policies pane, then linking the policy to the application definition by editing the application script and adding the RestrictVariable command. For more information on linking a password policy see “Linking a Policy to an Application” on page 63 .

Password policies are comprised of one or more password rules applicable to one or more single sign-on enabled applications and to specific directory objects. You can configure password policies in the **Password Policy Properties tables** of the Administrative Management utilities.

SecureLogin remembers the passwords and handles password changes after they expire on the back-end application. For example, after 30 days or when users decide to change their password. The SecureLogin password management functionality includes the capability to set password expiry duration and generate passwords that comply with specified password policies.

Password policies are typically created to match existing password policies. You should consult application owners before changing an existing password policy.

To determine the requirements and parameters of the password policy and the applications the password policy applies to, test complex policies on a test user account to ensure that they are viable.

Using Application Definition Wizard to Create Password Policy

You can create password policies through the application definition wizard while enabling application for single sign-on. For details on using the application definition wizard, refer to the *NetIQ SecureLogin Application Definition Wizard Administration Guide* .

However, you cannot use the wizard to edit or delete password policies.

Password Policy Properties

Organizations and applications often have rules about the content of passwords, including the required number and type of characters. The **Password Policy Properties** table helps you to create and enforce these password rules through a password policy, and apply this policy to one or more applications.

Table 7-1 *The Password Policy Properties Table*

Policy	Value To Be provided	Description
Minimum length	Whole number	Defines the required minimum number of characters.
Maximum length	Whole number	Defines the required maximum number of characters.
Minimum punctuation characters	Punctuation characters	Defines the minimum number of punctuation characters allowed in a password.
Maximum punctuation characters	Punctuation characters	Defines the maximum number of punctuation characters allowed in a password.
Minimum uppercase characters	Whole number	Defines the minimum number of uppercase characters allowed in a password.
Maximum uppercase characters	Whole number	Defines the maximum number of uppercase characters allowed in a password.
Minimum lowercase characters	Whole number	Defines the minimum number of lowercase characters allowed in a password.
Maximum lowercase characters	Whole number	Defines the maximum number of lowercase characters allowed in a password.
Minimum numeric characters	Whole number	Defines the minimum number of numeric characters allowed in a password.
Maximum numeric characters	Whole number	Defines the maximum number of numeric characters allowed in a password.
Disallow repeat characters	**No**/**Yes**/**Yes, case insensitive**	Disallows the use of repeated characters, or the use of the same successive characters. If this option is set to **No**, characters can be repeated. This is the default value. If this option is set to **Yes**, same alphabetic characters in a different case are considered as different characters. For example, A and a are different. If this option is set to **Yes, case insensitive**, the successive use of the same alphabetic characters in a different case is not allowed.

Policy	Value To Be provided	Description
Disallow duplicate characters	**No**/**Yes**/**Yes, case insensitive**	Disallows the use of the same non-successive characters. If this option is set to **No**, duplicate characters are allowed. This is the default value. If this option is set to **Yes**, the same alphabetic characters in a different case are considered as different characters. For example, A (uppercase) and a (lowercase) are different. If this option is set to **Yes, case insensitive**, duplication of the same alphabetic characters in a different case is not allowed.
Disallow sequential characters	**No**/**Yes**/**Yes, case insensitive**	Disallows the use of successive characters in an alphabetical order. If this option is set to **No**, sequential characters are allowed. This is the default value. If this option is set to **Yes**, sequential characters in a different case are considered as non-sequential. For example, a and b and non-sequential. If this option is set to **Yes, case insensitive**, sequential characters in different cases is disallowed.
Begin with an uppercase character	**No**/**Yes**	This parameter requires a password to start with an uppercase character. The default value is **No**. If this option is set to **Yes**, all other policies that indicate that a password must begin with a particular character or in a specific manner are disabled. **IMPORTANT:** Only one type of character can be designated as the first value of a password.
End with an uppercase character	**No**/**Yes**	Enforces the use of an uppercase letter at the end of a password. The default value is **No**. If this option is set to **Yes**, all other policies that indicate that a password must end with a particular character or in a specific manner are disabled.
Prohibited characters	Keyboard characters	Defines a list of characters that cannot be used in a password. **NOTE:** There is no need of a separator in the list of prohibited characters. For example, @#$%&*

Policy	Value To Be provided	Description
Begin with any Alpha character	**No**/**Yes**	This parameter requires a password to start with an alphabetical letter that is, [a-z] or [A-Z]. The default value is **No**. If this option is set to **Yes**, it automatically disables all other policies that specify what the first character of the password should be.
Begin with any number	**No**/**Yes**	Enforces the use of a numeric character as the first character of the password. The default value is **No**. If this option is set to **Yes**, it automatically disables all other policies that specify what the first character of the password should be.
Begin with any symbol	**No**/**Yes**	This parameter requires a password to start with a symbol. These characters are: ~!@#$%^&*()_+|-=\\{}[]:\\";'<>?/,.` The default value is **No**. If this option is set to **Yes**, it automatically disables all other policies that specify what the first character of the password should be.
End with any Alpha character	**No**/**Yes**	Enforces the use of an alphabetic character as the last character of the password. The default value is **No**. If this option is set to **Yes**, it automatically disables all other policies that specify what the password should end with.
End with any number	**No**/**Yes**	Enforces the use of a numeric character as the last character of the password. The default value is **No**. If this option is set to **Yes**, it automatically disables all other policies that specify what the password should end with.
End with any symbol	**No**/**Yes**	This parameter requires a password to end with a symbol. These characters are: ~!@#$%^&*()_+|-=\\{}[]:\\";'<>?/,.` The default value is **No**. If this option is set to **Yes**, it automatically disables all other policies that specify what the password should end with.

Creating a New Password Policy

To create a new password policy:

1. Access the Administration Management Utility.

   For information on accessing the Administrative Management utility, see “Accessing iManager”in the *NetIQ SecureLogin Installation Guide* and, or, “iManager Plug-In”.
2. Click **Password Policies**. The Password Policies page is displayed.
3. Click **New**. The New Password Policy dialog box is displayed.

   It is important to use a unique name for all logins, applications, and password policies. Password policies cannot have the same name as any other SecureLogin attribute. Typically, organizations employ the naming convention ApplicationNamePwdPolicy, for example, LotusNotesPwdPolicy.
4. In the **Enter a name for the new password policy** field, specify a name for the policy. The new policy is added under the Password Policies.
5. Click **OK**. The new password policy is added.
6. Click the new password policy. The Password policy properties table is displayed.

   The table contains **Description** and **Value** columns. Most policy rules are not enforced and do not have a default value. Values are either **Yes**, **No**, or a whole number.
7. In the **Description** column, locate the policy you want to change, then either select the appropriate value from the drop-down list or enter the required text field values in the adjacent **Value** column.
8. Click **Apply** to save changes.
9. Click **OK** to close the Administrative Management utility.

IMPORTANT: Password policies are linked to applications by using the SecureLogin application definition command `RestrictVariable`. You can use this command to apply password policies to one or more applications.

Linking a Policy to an Application

You can set or select a password policy, while enabling applications for single sign-on using the application definition wizard. You can also link the password policies to applications by using the SecureLogin application definition command `RestrictVariable`. With this command, you can apply the password policies to one or more applications, as in the example below.

The following definition restricts the `$Password` variable to the Finance password policy. The user’s password must match the policy when he or she saves the credentials. When the password requires changing, the application generates a new password based on the policy randomly because Random is included in the definition at ChangePassword.

```
# Set the Password to use the Finance Password Policy
RestrictVariable $Password FinancePwdPolicy
# Login Dialog Box
Dialog
   Class #32770
   Title “Login”
EndDialog
Type $Username #1001
Type $Password #1002
# Change Password Dialog Box
Dialog
   Class #32770
   Title "Change Password"
EndDialog
Type $Username #1015
Type $Password #1004
ChangePassword $Password Random
Type $Password #1005
Type $Password #1006
Click #1
```

The following example uses an application definition to restrict the `?NewPwd` variable to the Finance password policy. The user’s current password ($Password) is saved and used when the application starts for the first time and prompts the user to enter the credentials. When the password expires, the password policy is enforced on any new password.

```
# Set the Password to use the Finance Password Policy
RestrictVariable ?NewPwd FinancePwdPolicy
# Log on Dialog Box
Dialog
Class #32770
Title "Log on"
EndDialog
Type $Username #1001
Type $Password #1002
Click #1
# Change Password Dialog Box
Dialog
Class #32770
Title "Change Password"
EndDialog
Type $Username #1015
Type $Password #1004
ChangePassword ?NewPwd Random
Type ?NewPwd #1005
Type ?NewPwd #1006
Set $Password ?NewPwd
Click #1
```

8 Managing Smart Card Integration

Network authentication is the verification of a user's login credentials before granting access to a network or operating system. Users typically authenticate to a network using one of the following methods:

- Password
- Biometric device (fingerprint or iris scan)
- Smart card and PIN
- Token

When a user authenticates successfully and the operating system loads, SecureLogin starts and manages the login credentials to the user's single sign-on-enabled applications.

If you want to enforce biometric, smart card, or token authentication at the application (or transaction) level, `AAVerify` can be used with SecureLogin to prompt the user to re-authenticate before SecureLogin retrieves their credentials and logs in to single sign-on enabled applications.

You can also integrate network authentication methods such as ActivIdentity's SCPL with SecureLogin to manage user's Windows login credentials (user name, password, and network selection). SCPL provides secure and convenient network log in by allowing a user to simply insert the smart card and enter the PIN to gain network access. SCPL retrieves the user's Windows username and password from the smartcard and automatically enters these into the Windows Graphical Identification and Authorization (GINA)/Credential Provider interface after a user enters his or her PIN.

The use of a smart card with SecureLogin is based on the enterprise preference to have users utilize a smart card to log on and store their single sign-on data or to encrypt their directory data using a Public Key Infrastructure (PKI).

To continue working with SecureLogin, you must manually add the entry and set the value to File.

The secondary store entry under HKLM/Protocom/SecureLogin/Security is deleted when the installer is modified to remove smart card support. To continue working with SecureLogin, you must manually add the entry and set the value to File.

If you are using smart card authentication for the Citrix login prompt, enter the smart card PIN manually, because the PIN is not cached for the Citrix server authentication.

This section provides information on the following:

How SecureLogin Uses Smart Cards

This section provides information on the following:

Prerequisites

SecureLogin supports ActivClient 6.x and 7.x, and Microsoft BaseCSP MiniDriver only. ActiveClient 6.2 is supported on 32-bit and 64-bit system on all platforms.

To enable smart card support with SecureLogin, the **Use smart card** option must be selected during installation, regardless of the administrator’s intended preferences for setting the SecureLogin security preference **Require smart card is present for SSO and administration operations**.

IMPORTANT: Contact NetIQ Support for information on other cryptographic service providers.

Refer “Installing SecureLogin for Smart Cards” on page 70 in the *NetIQ SecureLogin Installation Guide* for more information on enabling smart card support during installation and deployment.

NOTE

When you use eDirectory to create a certificate for a smart card user, ensure that the key usage options **Digital Signature** and **Key Encipherment** are checked.

Using Smart Card to Log In to Workstation

SecureLogin allows a user to alternate their log in method by using smart card.

However, a user can only log in by using a smart card to access the SecureLogin credentials only if the smart card option is selected during installation.

If the smart card option is not selected during installation, a user attempting to access SecureLogin on the workstation is forced to log in with his or her username and network password.

Strong Authentication Methods

The following sections explain the strong authentication methods used in SecureLogin.

Advanced Authentication

SecureLogin uses the AAVerify script command to enforce strong security for applications that cannot provide such a mechanism natively. AAVerify can also be implemented to provide user authentication to applications that have no existing authentication interface. Use this command in conjunction with NetIQ Advanced Authentication Framework or NetIQ Modular Authentication Services (NMAS) to force users to log in to the configured application with a smartcard.

For details of the `AAVerify` application definition command, see the *NetIQ SecureLogin Application Definition Guide*.

New Functionality in the AAVerify Command

The existing version of the AAVerify command relies on NetIQ Modular Authentication Services (NMAS). Any NMAS supported method like smartcard, can be deployed at the backend to process any re-authentication requests. In Active Directory environments similar support is provided by NetIQ Advanced Authentication Framework.

The new AAVerify command was developed to specifically provide a secure method to re-authenticate a user. Thus proving the users identity before injecting the SecureLogin credentials into sensitive applications. In an enterprise or corporate environment, a sensitive application is one where a SecureLogin application definition is applied that calls for re-authentication.

To process the reauthentication request, the new `AAVerify` command now takes into account the method by which users are currently logged in, as well as their directory connectivity status.

If users have logged in with a username and password, they are prompted to reauthenticate by using the password, regardless of whether they are offline or online.

If users have logged in with a smart card, they are prompted to reauthenticate by using the original smart card PIN, regardless of whether they are offline or online.

The new `AAVerify` command is independent of NMAS and can be used to enforce strong user-friendly re-authentication by using a smart card and PIN or password without installing NMAS.

The new `AAVerify` command caters to a mixed environment where either of the following conditions exists:

- A user might log in to a number of workstations by using a combination of both smart card or password authentication
- A scenario where several users might log in to one workstation by either smart card or password authentication.

The New ?IsPin Variable

`?IsPin` is a new SecureLogin variable available in Microsoft Active Directory mode only.

The `?IsPin` variable is automatically generated when a user logs in and stores, information based on whether the user has logged in to the workstation by using a smart card and PIN, or has logged in by using a password.

When the `?IsPin` variable is called from an application definition, it indicates the following:

- If the returned value is true, it means that the user has logged in by using a smart card, and only the PIN value is passed through to the SecureLogin.
- If the returned value is false, it means that the user has logged with a password.

NOTE: The `?IsPin` variable is updated only at a login and is not updated at a screen unlock.

Recommended Configuration

The **Use smart card option** option is normally based on your preference to have the SecureLogin users utilize a smart card to store the single sign-on data or to encrypt their user's directory data by using a Public Key Infrastructure (PKI).

If you decide to allow users to log in to their workstations by using a smart card and reauthenticate against their smart card, then the **Use smart card option** option must be selected during the installation regardless of the option set for **Require smart card is present for SSO and administration operations**.

NOTE: We recommend that you use a smart card configuration policy to lock the screen on card removal to ensure that the smart card belongs to the currently logged-in user.

Example Application Definition

The following application definition shows how to call the `AAVerify` command based on the login method. It uses the Notepad application. After the Notepad application is started, the `AAVerify` command is invoked to prompt the user to reauthenticate, using the login method for the workstation.

```
Dialog
Class Notepad
EndDialog

OnException AAVerifyFailed Call AAVerifyFailed
OnException AAVerifyCancelled Call AAVerifyCancelled

If ?isPin Eq “true”
  AAVerify -method "smartcard" ?result
Else
  AAVerify -method "password" ?result
EndIf
ClearException AAVerifyFailed
ClearException AAVerifyCancelled
```

```
Type $username
Type \n
Type $password
Type \n
Sub AAVerifyFailed
   MessageBox "Reauthentication failed."
   EndScript
EndSub

Sub AAVerifyCancelled
MessageBox "Reauthentication cancelled."
EndScript
EndSub
## EndSection: "Login Window"
```

Reauthenticating a Predefined Web Application

If the new `AAVerify` command is used to reauthenticate a Web browser-based application or if the **Prompt for device authentication for this device** option is enabled for Web applications, then the predefined application definition for the Web browser must be applied for that particular user to avoid confusion when prompting for reauthentication.

One Time Password

The use of multiple passwords places high maintenance overheads on large enterprises. This results in significant cost where users use and manage multiple logins. The calls to helpdesk to reset forgotten password, providing all password when a new employee joins, or deleting the logins when an employee quits can be high in cost.

A one time password (OTP) reduces the cost, particularly with regard to calls to the help desk to reset a forgotten password, or to ensure that all passwords are provisioned when a new user starts, or deleted when existing user leaves the organization.

SecureLogin integrates with ActivIdentity's one time password authentication functionality and provides you access to the `GenerateOTP` application definition command, which can be used to generate synchronous authentication and asynchronous authentication soft token support for smart card user authentication.

If you are using One Time Password capability on 32-bit applications running on 64-bit operating system in the Active Client 7.x environment, set the following registry keys:

Location	Type	Name
HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\Security\	String	StorageDeviceInterfaceLibraryPKCS11-Wow64

Set to:	Explanation
C:\Program Files (x86)\HID Global\ActivClient\acpkcs211.dll	Path to ActivClient

Location	Type	Name
HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\Security\	String	StorageDeviceInterfaceLibraryPKCS11-2

Set to:	Explanation
C:\Program Files\HID Global\ActivClient\acpkcs211.dll	Path to ActivClient

Location	Type	Name
HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\Security\	String	StorageDeviceInterfaceLibraryPKCS11

Set to:	Explanation
C:\Program Files\HID Global\ActivClient\acpkcs211.dll	Path to ActivClient

Smart Card Password Login

ActivIdentity's Smart Card Password Login (SCPL) provides smart card-based Windows login that is not PKI-based. SCPL, when used in conjunction with SecureLogin, stores and manages a user's Windows login and SecureLogin credentials. It provides efficient network login by allowing a user to simply insert their smart card and enter their PIN.

Smartcard Application Reauthentication

You can configure SecureLogin to reauthenticate an application using the SecureLogin Administrative Management Utility or application definition wizard. To use this, enable **Prompt for device re-authentication for this application** and configure the **Re-authentication method**.

IMPORTANT: `Basecsp.dll` supports only smart card authentication. It does not support smart card re-authentication even in Kiosk mode. To allow re-authentication, use `acpkcs.dll`.

For more details refer Chapter 10, "Reauthenticating Applications," on page 91.

Installing SecureLogin for Smart Cards

Client Setup

During the installation of SecureLogin the smart card option can be selected by the administrator to enable a SecureLogin user to utilize a smart card to encrypt their directory data using a Public Key Infrastructure (PKI) token.

Existing ActivClient smart card settings are used by SecureLogin if they are detected unless the administrator chooses otherwise.

The administrator can optionally select an alternative cryptographic service provider (Microsoft Crypto API) from a drop-down list. SecureLogin supports ActivClient, and Microsoft BaseCSP MiniDriver smart card middleware. Contact NetIQ Support if your organization uses any other cryptographic service provider.

Server Side Administration Preferences

SecureLogin is a highly configurable and flexible product, with numerous preferences and options, that allow the system administrator to implement and enforce corporate directory policy across an enterprise.

Corporate policies may include, but are not limited to, enabling strong application security, how SSO data is encrypted and stored, how password and passphrase policies are implemented and enforced, and setting of management procedures for lost smart card scenarios.

In the case of strong security requirements, administrators should be fully aware of the implications of linking the use of SecureLogin to a smart card and disabling the passphrase functionality.

Various combinations and permutations of configuring SecureLogin for use with smart cards are covered in following sections.

Minimum Requirements

For general information about the minimum requirements for using smart cards with SecureLogin, see the *NetIQ SecureLogin Installation Guide*for your directory environment.

Supported Configurations

SecureLogin supports the following smart card middleware:

- ActivClient 6.x and 7.x
- Microsoft BaseCSP MiniDriver

NOTE: SecureLogin might work with other smart card vendor middleware but those are not tested and are not supported.

While installing SecureLogin with smart card option selected, select the appropriate cryptographic service provider and PKCS#11 dynamic link library file path. If the appropriate version of PKCS#11 library file is not present during installation, SecureLogin installs without smart card support. However, if a required library file is missing errors can occur.

For example, if the PKCS#11 wrapper library file `aetpksse.dll` is missing, the error message `Access to smart card failed` is shown when the Access Manager attempts to access the smart card. To avoid this error, ensure that the `aetpksse.dll` file is available at `C:\WINDOWS\system32\`.

PKCS 11 Library Path

Smart Card Middleware	PKCS 11 Library path
ActivClient 6.2	`C:\Program Files\ActivIdentity\ActivClient\acpkcs211.dll`
ActivClient versions previous to 6.2	`C:\Windows\Sytem32\acpkcs211.dll`

If smart card middleware is installed after SecureLogin is installed, the registry key settings for cryptographic service provider and PKCS#11 dynamic link library file path must be changed manually; to activate smart the card support, uninstall or re-install SecureLogin.

NOTE: Manually configuring a third party smart card PKCS #11 link library assumes a high level of understanding of the crypto-graphic service provider's product. System administrators are encouraged to use the ActivClient smart card support with SecureLogin whenever possible.

For detailed instructions about installing SecureLogin for use with smart cards and cryptographic tokens, see the *NetIQ SecureLogin Installation Guide*for your directory environment.

Configuring SecureLogin for Smart Cards

No two organizations have the same environment and requirements, SecureLogin includes a number of options that determine SecureLogin's behavior, such as how single sign-on data is encrypted (that is, using the smart card or a passphrase question and answer) and how to handle scenarios such as lost cards.

To configure the preferences, use the iManager in eDirectory environments, MMC plug-in for Active Directory environments, and SecureLogin Manager in LDAP v3-compliant directories such as Sun, Oracle, and IBM.

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Preferences**. The Preferences Properties table is displayed.
3. In the **Setting Description** column, go to **Security** and select the appropriate preferences.
4. Click **Apply**.
5. Click **OK**.

The following sections explain the various security preferences:

Using AES for SSO Data Encryption

This option determines the level and standard of encryption used to encrypt single sign-on data stored on the smart card by allowing the use of AES instead of triple DES.

If you select **No**, a 168-bit key used with triple DES (EDE) in Cipher-Block Chaining (CBC) mode is used to encrypt the user's single sign-on credentials.

NOTE: The input key for DES is 64 bits long and includes 8 parity bits. These 8 parity bits are not used during the encryption process, resulting in a DES encryption key length of 56 bits. Therefore, the key strength for Triple DES is actually 168 bits.

If you select **Yes**, then a 256-bit key used with AES (EDE) in CBC mode is used to encrypt the user's credentials.

If a previous version of SecureLogin has been implemented with passphrases enabled and if this option is set to **Yes**, users must answer with a passphrase before data can be decrypted and reencrypted by using AES.

Using PKI Encryption for the Datastore and Cache

If PKI credentials are used to encrypt SecureLogin data with the passphrase security system off (set to **No**), you should consider implementing a key archive/backup and recovery. If key archive/backup and recovery is not implemented and the passphrases security system is not enabled, the users can never decrypt their SecureLogin data if they lose their smart card because the private key is stored on the lost smart card.

Without private key recovery, you have to clear the user's SecureLogin data store before they can use SecureLogin again. This is a high security solution but is inconvenient to end users as they cannot access SecureLogin without the smart card.

- "Choosing a Certificate" on page 73
- "Certificate Selection Criteria" on page 74
- "Current Certificate" on page 74

Choosing a Certificate

When a smart card is configured to use PKI credentials to encrypt single sign-on data, SecureLogin retrieves the serial number of the current certificate and locates the certificate in the certificate store as specified in the relevant SecureLogin preferences. SecureLogin then loads the associated private key and attempts to decrypt the user key with the private key.

If the decryption fails or the certificate is not located, a smart card is present, and a certificate that matches the selection criteria is not located, then SecureLogin assumes that a recovered smart card is in use. It then attempts to decrypt the user key with each key pair stored on the card.

IMPORTANT: If you are using PKI encryption and the certificate selection criteria depends on the certificate's friendly name, you will need to disable Microsoft certificate propagation.

Because the windows certificate propogation method does not propagate the certificate friendly name, you cannot successfully start SecureLogin.

To disable the Microsoft certificate propogation, set the registry key value to 0.

1. On the Windows **Start** menu, click **Start > Run** to display the Run dialog box.
2. Type `regedit` then click **OK** to open the Registry Editor.
3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\ScCertProp`
4. Create a DWORD Value named Enabled.
5. Set the value of the DWORD to 0.
6. Exit the Registry Editor.

Certificate Selection Criteria

The Certificate Selection Criteria preference allows you to select an encryption or authentication certificate to encrypt the user's single sign-on information in the directory.

The certificate selection criteria determines which certificate to select if multiple certificates are in use (for example, if an enterprise has configured an Entrust certificate for single sign-on encryption and a Microsoft certificate for login and or, authentication).

If only one certificate is used, the field is blank and the certificate is detected automatically and set to **User Certificate**. When entering certificate selection criteria, no special formatting is required and the search string is not case sensitive. Wildcards are not used and a search matches if the search text is a substring of the certificate subject field. SecureLogin attempts to match against certificate subject, issuer, and friendly name in the following order:

1. Certificate Subject
2. Certificate Issuer
3. Friendly Name

Example 8-1 *For example if the subject is*

```
CN=Writer,OU=Users,OU=Accounts,OU=APAC,DC=Novell,DC=Int
```

Then Writer is a valid search value, as are *Accounts*, *APAC*, and *Int*. The prefixes CN=, OU=, or DC= are not required.

Similarly, if the **Certificate Issuer** is

```
CN=IssuingCA1,OU=AD,DC=undiscovered,DC=com
```

Then *IssuingCA1* is a valid search value, as are *AD*, *undiscovered*, and *com*.

Current Certificate

This preference displays the certificate that is currently being used by SecureLogin to encrypt a user's single sign-on data.

Lost Card Scenarios

Lost Card Scenario Preference

The Lost Card Scenario preference determines how SecureLogin handles a user forgetting, losing, or damaging a smart card. The Lost Card scenario preference can only be used if, the Enable passphrase security system preference is also enabled (set to either Yes or Hidden).

NOTE: For users upgrading from SecureLogin version 5.5, setting **Enable passphrase security system** to **Hidden** is equivalent to setting the old **Disable passphrase security system** to **Off**.

If a smart card is used to encrypt single sign-on data and the card is lost, stolen, or damaged. Also, if the key archive/backup and recovery is not used, the user will not have access to their single sign-on data unless the Enable passphrase security system preference is set to Yes or Hidden.

- If Enable passphrase security system is set to Yes, the user has previously set a passphrase, and Lost card scenario is set to Allow Passphrase, then the user is prompted to answer with his or her passphrase before SecureLogin is available.
- If **Enable passphrase security system** is set to **Hidden**, the user is not prompted for the answer and SecureLogin loads seamlessly.

Allow Passphrase

This preference allows the user to start SecureLogin using their passphrase if their smart card is not available. The **Enable passphrase security system** preference must be set to Yes or Hidden for this to work. Hidden replaces a user-generated passphrase with a system-generated passphrase, effectively removing the need for the user to remember the passphrase answer.

IMPORTANT: For the user to decrypt data using their passphrase, the passphrase must already have been set. Administrators cannot simply toggle the **Enable passphrase security system** preference on the day the user forgets their smart card unless the user has previously set a passphrase (or had it randomly generated using Hidden).

NOTE: Administrators can manually disable inheritance of higher level preferences by selecting the Yes option for **Stop walking here** in the SecureLogin Administrative Management Utility, Preferences – General options.

Default

The default preference is to allow the user to start SecureLogin using their passphrase, unless it inherits a **Lost card scenario** preference from a higher-level container.

Requiring a Smart Card

The **Require smart card** preference prevents a user from starting single sign-on without his or her smart card. This option is for high security implementations where organizations want to tie the use of a user's single sign-on credentials to the user's smart card. This means that the user cannot access single sign-on with any other method; that is, they cannot use a username and password without the smart card.

IMPORTANT: If the **Require smart card** option is changed while the user is logged in, refreshing the cache using the **Advanced** > **Refresh Cache** option from the taskbar does not refresh the **Lost card scenario** option.

The user must log out and log in again (or restart SecureLogin) for the new option to take effect.

Allowing a Passphrase

The **Allow passphrase** preference must be used in conjunction with the **Enable passphrase security system** option. It allows the user to start SecureLogin by using a passphrase if the smart card is not available. The passphrase security system must be set to **Yes** or **Hidden** for this setting to apply.

The **Hidden** option replaces a user-generated passphrase with a system-generated passphrase, effectively removing the need for the user to remember the passphrase answer.

IMPORTANT: For the user to decrypt data using a passphrase, the passphrase must already be set. You cannot simply toggle the **Enable passphrase security system setting** to on the day the user forgets a smart card unless the user has previously set a passphrase (or had it randomly generated by using the **Hidden** option).

The **Default** option allows the user to start SecureLogin by using a passphrase if the smart card is not available through the **Allow Passphrase** preference. Alternatively, this option inherits the **Lost Card scenario** preference set by the higher-level container.

You can manually disable inheritance of higher-level options by selecting the **Yes** option for **Stop walking here** (SecureLogin Administrative Management utility > **Preferences** > **General** options.)

Passphrases for Temporary Access

There is another option available that permit access if a user loses or forgets his or her smart card. For example, If a user loses or forgets his or her smart card and the **Lost card scenario** option is set to **Require smart card**, you can grant temporary access to systems by resetting the user's password. The user is then required to log in and enter the passphrase. This option is possible only if the **Enable passphrase security system** is turned on.

However, the user should not expect easy or automatic access to the system. Users should understand that, a strong and secure solution has been implemented and that they have the responsibility of looking after their own smart cards.

Using a Card Management System

Enterprise server or web-based card management system (CMS) software enables corporations to implement and easily manage smart card-based identity management, provisioning, and authentication devices and enforce policy across geographically-dispersed locations.

These systems provide a complete and flexible solution to manage the issuance, administration, and configuration required for the successful and seamless smart card integration.

NetIQ CMS provides a complete and flexible solution to manage the issuance, administration and configuration required for a successful and seamless smart card integration with SecureLogin and Smart Card Password Login (SCPL). It can be configured to perform key escrow, archive and recovery as described throughout this document.

- “Restoring a Smart Card Using Card Management System” on page 77
- “Accessing Without a Card Management System” on page 77
- “PKI Credentials” on page 78
- “Key Generated on Smart Card” on page 78

Restoring a Smart Card Using Card Management System

The use of a CMS is crucial if an enterprise opts to deploy corporate smart cards with a very high level of security. This would include disabling the Enable passphrase security system preference combined with Store credentials on smart card (set to Yes) and Use smart card to encrypt SSO data (set to PKI credentials or Key generated on smart card).

In the event of a lost or damaged smart card, the user can never decrypt their single sign-on data because the key stored on the smart card is not recoverable.

You must then reset the user's corporate passwords and issue a new smart card (with a new key pair) before the user can log in and reconfigure the single sign-on applications using SecureLogin again.

The user must manually enter all application credentials into SecureLogin the first time he or she logs in after the data was cleared from the directory.

Enterprises should consider implementing key escrow, archiving, or backup through a suitable CMS to allow a user's encryption key to be recovered in the event of a lost or damaged smart card.

It is recommended that you extensively test the CMS and smart card restoration techniques before selecting the high security options described above.

The procedure to reset a user’s data store is described in “Deleting or Re-setting User Data” on page 20.

Accessing Without a Card Management System

If an enterprise opts to deploy corporate smart cards without a suitable card management system (CMS), you can still create a very high level of security by setting Enable passphrase security system to No and then selecting the Use smart card to encrypt SSO data preference to PKI credentials or Key generated on smart card. However, in the event of a lost or damaged smart card the user can never decrypt the single sign-on data because the key stored on the smart card is not recoverable.

WARNING: Deleting the user's single sign-on datastore permanently deletes the users data in the directory. This would include any local applications, credentials, preferences, and password policies that are not being inherited from some other object in the directory.

If you still decide to delete the user's existing single sign-on configuration data store, delete it from the Advanced Setting > Datastore tab of the user object in the directory management tool.

The administrator must then reset the user's corporate password and issue a new smart card (with a new key pair) before the user can log on and reconfigure their single sign-on enabled applications.

The user will have to re-enter all their application credentials into SecureLogin the first time it is used after having them deleted from the directory.

PKI Credentials

If the Use smart card to encrypt SSO data preference is set to PKI credentials and Enable passphrase security system is set to No, then in the event of a lost or damaged smart card the user will never be able to decrypt the single sign-on data because the key stored on the smart card is the only key that can be used for decryption. This key is not recoverable unless key archiving and recovery was implemented.

If a CMS-based key archive is used, then the encryption key needs to be recovered to the new smart card, the single sign-on data unencrypted, and an administrator needs to chose a new certificate to encrypt the user's data.

If you are using the enterprise CMS-based recovery system, you must issue the user a replacement smart card based on a CMS backup of the user's original key.

Key Generated on Smart Card

Similarly, if the Use smart card to encrypt SSO data preference is set to use Key generated on smart card, then in the event of a lost or damaged smart card the user can never decrypt the single sign-on data because the key stored on the smart card is not recoverable.

You should consider setting the **Enable passphrase security system** option to **Yes** when the **Key generated on smart card** option is used to provide an alternative mechanism for decrypting single sign-on data if the smart card is lost/stolen/damaged.

Using the enterprise CMS-based recovery system, the administrator must issue the user a replacement smart card based on a CMS backup of the user's original key. The replacement card includes the recovered private key and a new key pair so data can be decrypted using the old key and re-encrypted using the new key.

Smart Card with DAS Integration

In the earlier versions of SecureLogin, Active Directory authentication of the workstation was used to log in to SecureLogin.With SecureLogin, you can enable users to log in to SecureLogin separately by using the smart card credentials.

With SecureLogin, you can enable users to log in to SecureLogin separately by using the smart card credentials

To enable this behavior:

1. On the Windows desktop, click **start > Run** to display the Run dialog box.
2. Enter `regedit,` then click **OK** to open the Registry Editor.
3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\NSLADAuth`.
4. Create `DWORD NSLADAuth` and set the value of `NSLADAuth` to 1.
5. Exit the Registry Editor.
6. Log out from the workstation and log in again.
7. Launch the SecureLogin.

This feature can be effectively used for Desktop sharing using DAS. To enable smart card with DAS you have to use `on-cardmon` and card-insert elements.

This feature enable users to log in by using the Smart Card credentials in Desktop Automation Services, the `on-cardmon` element is modified. The `card-insert` and `LoginAction` attributes are added to the `on-cardmon` element. For more information on these new attributes, see "on-cardmon" on page 140 in Administering Desktop Automation Services.

Fast user switch using Smart Card in Active Directory Mode

The changes in smart card and Desktop Automation Services allow for switching of users using a smart card in Active Directory mode.

1. Log in to the workstation where you want to launch SecureLogin.
2. Configure `actions.xml` to hide and unhide the desktop using the smart card fast user switch.
3. Insert the smart card.

   The SecureLogin PinPrompt dialog box is displayed prompting you to enter a valid PIN.
4. Enter the valid PIN.

   SecureLogin is launched successfully for the smart card user.

Disconnected Login using NESCM

With this feature enabled, LDAPAuth will encrypt and store the Windows workstation or the AD domain user password locally and retrieve it when required, hence the user need not re-enter the workstation password from the next login onwards.

With NESCM as the login method, this feature is supported in the LDAP Gina or Credential Provider mode with eDirectory. When logging in with NESCM in either online or offline mode, just entering the PIN is sufficient.

Registry Settings

To enable Disconnected Login using NESCM, create the following registry settings:

Registry Path	Registry Type	Registry Name	Registry Value
`HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Login\LDAP`	DWORD	`DisconnectedRequired`	`1`
	DWORD	`LDAPAuthNMASSelected`	`1`
	DWORD	`UsePasswordFieldforNMAS`	`1`
	String	`LDAPAuthNMASSequence`	`Enhanced Smart Card`
	DWORD	`DoNotShutdownNSL`	`1`
`HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin`	DWORD	`TryRegCredInOffline`	`1`

SecureLogin in Kiosk Mode

In Active Directory mode, by default SecureLogin uses the workstation logged in session to login to SecureLogin. User can also login to SecureLogin using different credentials by updating the registry setting:

1. On the Windows desktop, click **start > Run** to display the Run dialog box.
2. Enter `regedit,` then click **OK** to open the Registry Editor.
3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\NSLADAuth`.
4. Create `DWORD NSLADAuth` and set the value of `NSLADAuth` to 1.
5. Exit the Registry Editor.
6. Log out from the workstation and log in again.
7. Launch the SecureLogin.

This feature can be effectively used for desktop sharing using DAS. To enable smart card with DAS you have to use `on-cardmon` and `card-insert` elements. For more information on these new attributes, see “on-cardmon” on page 140 in Administering Desktop Automation Services.

NOTE: This feature is not supported with SecureLogin installed in ADAM mode

---

Kiosk Mode using Smart Card

The changes in smart card and Desktop Automation Services allows switching of users using smart card in Active Directory mode.

1. Log in to the workstation where you want to launch SecureLogin.
2. Insert the smart card and Launch SecureLogin.

   The **NSL PinPrompt** dialog box is displayed prompting you to enter a valid PIN.
3. Enter the valid PIN.

   The SecureLogin is launched successfully for the smart card user.

Kiosk Mode without using Smart Card

If the user's smart card is invalid or not present, then they can login to SecureLogin with a different credential.

1. Log in to the workstation and launch the SecureLogin.
2. The **NSL Login** dialog box is displayed prompting you to enter a valid username and password.
3. Enter the valid credentials

   The SecureLogin client is launched successfully.

NOTE: If the Active Directory user password is expired, user has to change the password through the Windows settings before logging in to Novel SecureLogin. In this case, SecureLogin will not prompt the user to change an expired password.

Enable Pin Caching for Smart Card

To enable pin caching for smart card, update the registry settings:

1. Browse to `HKEY_LOCAL_MACHINE\Software\Protocom\SecureLogin`.
2. Create a new `DWORD` value `EnableSmartCardPinCache`.
3. Set this value to `1`.

Changing Smart Card Login Password on Expiry

In eDirectory mode when the user logs in using NESCM (Novell Enhanced Smart Card Method) and the user password is expired, SecureLogin detects the expired password and changes automatically on behalf of the logged in user. To enable this, update the registry settings:

1. Browse to `HKEY_LOCAL_MACHINE\Software\Protocom\SecureLogin`.
2. Create a new `DWORD` value `ChangePasswordOnExpiry`.
3. Set this value to `1`.

NOTE: This feature is supported only with SecureLogin installed in eDirectory Novell Client mode without selecting **Novell SecretStore Client**.

The `ChangePasswordOnExpiry` feature does not work with complex eDirectory password policy settings in SecureLogin.

---

9 Enabling Terminal Emulator Applications

This section provides information on enabling applications for single sign-on, enabling terminal emulator applications, and single sign-on support for MEDITECH applications.

It consists of the following sections:

For detailed explanation on enabling single sign-on for Windows, Web, and Java applications, refer the *NetIQ SecureLogin Application Definition Wizard Administration Guide* .

Enabling Terminal Emulator Applications

You can configure terminal emulators for single sign-on in the application definition editor in the Administrative Management utility, in the SecureLogin Client Utility, and the Terminal Launcher tool.

To enable a terminal emulator for single sign-on, you must run the terminal launcher application (tlaunch.exe), and link the emulator definition to the terminal emulator application definition.

Terminal Launcher allows you to configure emulator definitions for use with SecureLogin.

The following sections document these procedures:

NOTE: Contact NetIQ Support for information on using a ViewNow terminal emulator.

In the following sections, we use Eicon Aviva. Although these procedures apply to most terminal emulators, the application definition and other configuration information might differ for each emulator application. Contact Support for help.

Creating and Saving a Terminal Emulator Session File

Prior to enabling any terminal emulator for single sign-on, you will need to include or create a session file that provides all the required settings for the server connection and any other parameters required for deployment to users. Terminal Launcher should be configured to run this session file when

launching the emulator. Any modifications to the session must be saved to this file. The session file can be saved locally or on the server. In most environments, the session file already exists and you only need to configure Terminal Launcher to point to the relevant file.

1. Start the terminal emulator application.
2. Connect to the required host.
3. Change the terminal emulator settings as required.
4. Save the session. The default directory is usually the application's installation directory.
5. On the **Connection** menu, click **Disconnect**.The session file remains loaded, but you have disconnected from the host.
6. On the **File** menu, click **Save** [session name] to save changes to the session file.
7. Exit the terminal emulator application.

Building a Terminal Emulator Application Definition

1. Open the SecureLogin Client Utility of SecureLogin by double-clicking , or by selecting **Start > Programs > NetIQ SecureLogin > NetIQ SecureLogin.**
2. Select **File > New > Application**. The New Application dialog box is displayed.
3. Select **New Application Definition**.
4. In the **Type** drop-down list, click **Terminal Emulator**.
5. In the **Name** field, specify a name for the application definition (in this example, Eicon Aviva), then click **OK**. The new application definition is added to the Applications pane.
6. Select the new application definition. The **Details** tab is displayed.
7. Click the **Definition** tab. The application definition editor is displayed.
8. Delete the default text displayed in the text box: `# place your application definition here`
9. In this example for Attachmate, type the following in the text box:

```
WaitForText "WELCOME TO ATTACHMATE"
Type @E
WaitForText "ENTER USERID -"
Type $Username
Type @E
WaitForText "Password  ===>"
Type $Password
Type @E
WaitForText " Welcome to Attachmate"
WaitForText "***"
Delay 1000
Type @E
```

   You must type the screen syntax accurately in the application definition editor; otherwise it will fail to operate. Wherever possible, cut and paste the text directly from the emulator screen into the editor.

10. Click the **Details** tab.
11. Ensure that the **Enabled** check box is selected.
12. Click **OK**.

Running a Terminal Launcher

Terminal applications are invoked by the terminal launcher when configured properly. After you create the application definition in the management utility, you must use Terminal launcher to link it with the appropriate emulator definition.

By setting tlaunch to point to the new application and the emulator definition you can click on the Create Shortcut button to create a shortcut that links everything together. When clicking on the shortcut tlaunch will launch the emulator with the configured session file. Once tlaunch can successfully communicate with the emulator it will then invoke the application definition to interact with the host session.

IMPORTANT: For successfully using Terminal Launcher on Windows XP SP3, the screen resolution must be 1024 by 768 pixels.

---

1. Select **Start > Programs > NetIQ SecureLogin > Terminal Launcher**. The Terminal Launcher dialog box is displayed.

   This release of NetIQ SecureLogin provides two Terminal Launcher shortcuts: one each for 32-bit and 64-bit. To launch **Terminal Launcher 32** or **Terminal Launcher 64**, click **Start** > **All Programs** > **NetIQ SecureLogin** > **Terminal Launcher 32** or **Terminal Launcher 64**.

   Use Terminal Emulator 32 to interact with 32-bit emulators. Use Terminal Emulator 64 to interact with 64-bit emulators.

2. In the **Available applications list**, click the required application definition (in this example, Eicon Aviva).
3. Click **Add** to move the selected application to the **Login to** list.
4. Click **Edit Available Emulators**. The Available Emulators dialog box is displayed.

   If you launch Terminal Launcher as a normal user on Microsoft Windows Vista or higher, the Edit Available Emulator button is dimmed. You must have administrator rights to edit the TLaunch.ini file. To edit the `TLaunch.ini` file:

   4a Click **Start** > **All Programs** > **NetIQ SecureLogin**, select **Terminal Emulator 32** or **Terminal Launcher 64**.

   4b Right-click on **Terminal Launcher 32** or **Terminal Launcher 64**, then select **Run as administrator**.

5. In the **Available Emulators** list, click on your emulator definition. In this example we clicked on `Eicon Aviva`.
6. Click **Edit**. The HLLAPI Emulator Configuration dialog box is displayed.
7. In the **Emulator Path** field, specify the emulator executable's location.
8. In the **Home Directory** field, specify the emulator's home directory.
9. In the **HLLAPI DLL** field, specify the file name and path.
10. In the **Session Files** field, select and delete the current session files.
11. Click **Add**. The Emulator Session File dialog box is displayed.
12. Browse and select the configured session file.
13. Click **OK** to close the Emulator Session File dialog box.
14. Click **OK** to close the HLLAPI Emulator Configuration dialog box.
15. Click **Done** to close the Available Emulators dialog box.

16 In the Terminal Launcher dialog box, ensure that Eicon Aviva is selected in the Emulator drop-down list.

17 Save the changes and click `Launch`.

You can choose to start emulator applications from within Terminal Launcher; however, users might not have access to Terminal Launcher or you might need to create multiple definitions for different hosts or sessions. To simplify this process for users, a desktop shortcut can be created by clicking on the Create Shortcut button.

It is important to understand that for SecureLogin to interact with the emulator, the emulator itself must be invoked by tlaunch. It is tlaunch that will launch the emulator and then invoke the NSL script to interact with the terminal session. If a user were to launch the emulator directly then tlaunch would not be running and no interaction would occur.

Creating a Terminal Emulator Desktop Shortcut

Do the following to create a shortcut for an application or a server:

1 Select **Start > Programs > NetIQ SecureLogin > Terminal Launcher**. The Terminal Launcher dialog box is displayed.

2 Click **Create Shortcut**. The Terminal Launcher Shortcut Options dialog box is displayed.

3 Select **Location > Desktop**.

4 Select the appropriate options from **Options.**

NOTE: **Quiet mode** and **Suppress errors** are the default options.

5 In the **Command Line** field, ensure that the following parameters are included (in this example, `/auto /e"Attachmate" /pAttachmate /q /s`):

Parameter	Description
`/auto`	Indicates to Terminal Launcher that the following is a parameter requesting the execution of a terminal emulator application that is configured for single sign-on. This parameter is mandatory.
`/e[application name]`	Initiates the execution of the terminal emulator.
`/p[Terminal Launcher config name]`	Initiates execution of the application created in Terminal Launcher.
`/q`	Quiet mode (no Cancel dialog box).
`/s`	Suppress errors.

6 Add additional parameters as required. For more information see, “Setting Terminal Launcher Command Line Parameters” on page 87

7 Click **Create**.

The shortcut is created on the desktop and you can deploy it to users in the preferred mode for your organization.

8 Click **Close** to close the Terminal Launcher dialog box.

9 Double-click the short cut.

The terminal emulator application is executed with Terminal Launcher and the Enter your credentials dialog box is displayed.

10 In the **Enter login credentials** fields, specify your username and password.

11 Click **OK**.

SecureLogin stores the user's login credentials and uses them to log on to the host through the emulator session. Subsequently, double-clicking the desktop shortcut logs the user directly on to the host without any further user interaction.

Setting Terminal Launcher Command Line Parameters

To run the required terminal emulator, Terminal Launcher command line parameters are included in the desktop shortcut command. For more information, see "Creating a Terminal Emulator Desktop Shortcut" on page 86.

The following table lists all of the available tlaunch parameters (also referred to as switches).

Table 9-1 *Terminal Launcher Command Line Parameters*

Parameter	Description
`/auto`	Indicates to Terminal Launcher that the following is a parameter requesting the execution of a terminal emulator application that is configured for single sign-on. For example: `C:\<....>\TLaunch.exe /auto /pApplication1` **NOTE:** This parameter is mandatory.
`/p[platform/application/Application Definition name]`	Initiates the execution of the terminal emulator as listed in the **Terminal Launcher Login to** field. To run multiple applications from the same command, add `/p[TL application/Application Definition name]` You can run up to fifteen applications simultaneously from the shortcut command line. For example: `C:\<....>\TLaunch.exe /auto /eAttachmate /pApplication1 /pApplication2` **NOTE:** You must type the emulator name exactly as it appears in the **Terminal Launcher Available Emulators** drop-down list.
`/b`	Specifies the background authentication mode.
`/c`	Allows the application to close in case of any errors. For example: `C:\<....>\TLaunch.exe /auto /c /Application` Application refers to the terminal emulator application configured for single sign-on.
`/e[emulator name]`	The parameter `/e[Terminal Launcher config name]` initiates the execution of the terminal emulator as listed in the **Terminal Launcher Available Emulators** drop-down list. **NOTE:** You must type the emulator name exactly as it appears in the **Terminal Launcher Available Emulators** drop-down list.

Parameter	Description
`/h[hllapi short name]`	Commands `TLaunch.exe` to connect to the specified HLLAPI session.
`/k[executable name]`	Quits (kills) the specified executable prior to launching the terminal emulator.
`/l`	Disables the terminal emulator execution. This parameter is required when multiple instances of terminal emulator are running and you want to disable a few selected instances.
`/m`	Enables multiple concurrent connections to specified sessions. This parameter is required for background authentication.
`/n`	Starts the selected terminal emulator without executing a SecureLogin application definition. For example: `C:\<...>\TLaunch.exe /auto /n` **NOTE:** This parameter does not function with VBA emulators. It overrides `/p` option.
`/n[number 1-15]`	Starts the specified number of terminal emulator sessions without executing SecureLogin application definition. For example: `C:\<...>\TLaunch.exe /auto /n3` **NOTE:** This parameter does not function with VBA emulators. It overrides `/p` option.
`/q`	**Quiet Mode** (no Cancel dialog bSox). For example: `C:\<...>\TLaunch.exe /auto /q`
`/s`	Suppress errors.
`/t`	Unlimited timeout during connection. For example: `C:\<...>\TLaunch.exe /auto /eAttachmate /pBackground /b /t /m /hA /s /q`
`/w`	Allows terminal emulator to wait for an application or process to complete before executing the next process.
`/x [Shared Access Rights]`	Setting EHLLAPI shared access for read and write permission between multiple EHLLAPI sessions. For example: `C:\<...>\TLaunch.exe /auto /eAttachmate /pBackground /b /t /m /hA/s /q /xSUPER_WRITE`

Parameter	Description

/hwnd	Use this parameter to pass an application handle to the terminal launcher. This parameter indicates the application window the terminal launcher should interact with.
	
	For example: This is an application definition script that uses GetHandle command to get the handle and passes it to TLaunch.exe using the /hwnd parameter.
	
	`GetReg "HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths\SLProto.exe\Path" ?SLLocation`
	`If ?SLLocation eq "<NOTSET>"`
	`EndScript`
	`EndIf`
	
	`GetHandle ?PuttyHWND`
	`Strcat ?TLaunch ?SLLocation "tlaunch.exe"`
	`Strcat ?TLaunchHWND "/hwnd" ?PuttyHWND`
	`Run ?TLaunch "/auto" "/ePutty" "/l" "/pPutty - Detection and Login" "/t" "/q" "/s" ?TLaunchHWND`

Table 9-2 *List of session options for EHLLAPI shared access support*

Parameter	Description
/xWRITE_WRITE	By default the application has write access and allows only supervisory application to concurrently connect to its presentation space (WRITE_SUPER). Using WRITE_WRITE parameter allows other applications with predictable behavior to share the presentation space. For example: C:\<....>\auto /e"Attachmate" /Tlaunch_script /q /s /xWRITE_WRITE
/xSUPER_WRITE	Allows other applications to have write access permissions, while the originating application may only perform supervisory functions.
/xWRITE_SUPER	Allows other applications to perform supervisory functions, while the originating application has complete control of the session.
/xWRITE_READ	Allows other applications that perform read-only functions to share their session, while the originating application has complete control of the session.
/xWRITE_NONE	Does not allow any other applications to share the presentation space.
/xREAD_WRITE	Allows other applications to write to the session, while the originating application has read-only access.

For more information, see Emulator Programming.

Support for the MEDITECH Predefined Application

SecureLogin supports MEDITECH 3.*x* and 4.*x*. It is dependant on the mandatory presence of the MEDITECH mrwscript.dll. The .dll file is provided by MEDITECH and must be installed during the installation of the MEDITECH application on the workstation.

NOTE: If you are an existing customer of Meditech, you can obtain the `mrwscript.dll` as part of your MEDITECH support agreement.

During the installation of the predefined MEDITECH application, SecureLogin detects the presence of the application and immediately warns, if the file cannot be located.

10 Reauthenticating Applications

With SecureLogin, when a user runs an application, SecureLogin seamlessly retrieves the user's application credentials and authenticates in the background so that the user is not prompted to specify the password. You can also configure SecureLogin to prompt the user for stronger authentication to all or specific applications.

Individual applications can be re-authenticated against any advanced device where SecureLogin is used. For example: NMAS™ infrastructure.

You can configure SecureLogin to request application reauthentication by using one of the following methods:

- "Using the Administrative Management Utility to Reauthenticate Applications" on page 91
- "Using the Application Definition Wizard to Reauthenticate Applications" on page 91
- "Using the AAVerify Command to Reauthenticate Applications" on page 92

NOTE: For environments that use the NMAS infrastructure, you can add the NMAS method in the **Reauthentication Method** (in the wizard login form definition screen) value by providing a free text string.

Using the Administrative Management Utility to Reauthenticate Applications

If you are using NMAS as the re-authentication component:

1. Launch SLManager.
2. Click **Applications**. The Application pane is displayed.
3. Double-click the application that you want to use for reauthentication.
4. Click the **Settings** tab. The Settings Properties table is displayed.
5. Set the value for **Prompt for device reauthentication for this application** to **Yes**.
6. From the **Reauthentication Method** drop-down list, select the device that you will use for reauthentication. Click **Any** if you want the user to choose from any of the available methods.

   NOTE: This option is not available through the iManager SSO plug-in

Using the Application Definition Wizard to Reauthenticate Applications

When creating an application definition to handle an application login screen through the application definition wizard, you can configure the definition to re-authenticate the user.

Refer the *NetIQ SecureLogin Application Definition Wizard Administration Guide* for detailed information.

Using the AAVerify Command to Reauthenticate Applications

For applications that cannot enforce user authentication natively, use the AAVerify command to enforce stronger application-based reauthentication such as biometric, token, or smartcard authentication.

It is recommended that the AAVerify command be placed in the application script just prior to the normal login section. This way the AAVerify command requests the pre-configured strong reauthentication method before SecureLogin retrieves and enters the username and password for the application.

For more information on the `AAVerify` command, read “AAVerify ” in the *NetIQ SecureLogin Application Definition Guide*.

11 Managing Application Definitions

This section provides information on the following:

Responding to Application Messages

When building an application definition for an application using the wizard, it is very important to respond appropriately to any message that the wizard displays. You must include actions for each of these messages in the application definition for SecureLogin to function correctly.

Responding to Login Notifications

A login notification is a message that the application displays after a user login has submitted. This message is meant to notify you of the status of the login request. For example, the login notification might indicate that you have successfully logged in or that you have specified a wrong password.

You can define how SecureLogin must handle login notifications in your application definition. Refer the for information on configuring SecureLogin to handle login notifications.

Adding Support for Password Changes

Depending on your organization's policies regarding password expiration, users might be required to change their passwords on a regular basis. Each time a user's password is changed for an application that is enabled for single sign-on, SecureLogin must update the stored credential information so that the next time the application is launched SecureLogin can provide the new password information.

To ensure that user password changes are updated in SecureLogin, it is important to configure SecureLogin to respond to the Change Password dialog box.

You can configure SecureLogin to automatically generate a new password (according to password policy, if required) whenever the Change Password dialog box is displayed. A randomly generated password is safer than a user-defined, reusable password.

Responding to Change Password Notification

A change password notification is a message that the application displays after a user has submitted a new password.

This might be either a confirmation or error message. This notification is important to SecureLogin because it provides the information necessary for SecureLogin to determine if it needs to update the stored credentials of the application.

If a password change notification is not defined SecureLogin prompts the user to define the notification, after changing the password.

12 Adding Multiple Logins

SecureLogin allows you to enable multiple logins for single sign-on to the same application. Before enabling your additional logins for single sign-on, make a list, including usernames and passwords, with a name to uniquely identify the login.

The following is an example list:

Table 12-1 *List of Additional Logins*

Name	User Name	Password
The default Application credential (typically the name of the application)		
Administrator	admin	123456
Support	help	abcdef
User	test1	xyz123

When the list is completed, use it to provide information as you complete the following procedure:

1. Enable the first account for single sign-on.
2. In the notification area (system tray), right-click the NetIQ SecureLogin icon, then select **New Login**. The Add New Login Wizard Welcome page is displayed.
3. Select the application for which you want to add another login. Let us consider, Gmail.
4. Click **Next**.
5. In the **Description** field, specify a descriptive name for the login. For example, Administrator.
6. Click **Finish**.
7. Start the application.

   The `[application]` login selection dialog box is displayed.
8. Select the required login credential set, then click **OK**.

   SecureLogin enters the credentials, and you are automatically logged on to the application.
9. The password value is left unassigned and the user will be prompted for the password when they select this login after executing the application.

13 Distributing Configurations

This section provides information on the following:

About Distributing Configurations

SecureLogin preferences, application definitions, password rules, and credentials are collectively the SecureLogin configured user environment. You can deploy and maintain this environment at all object levels, including import/export by file, copy to another object, etc. SecureLogin data can be added to users, containers, groups and even through Group Policy Objects in Active Directory environments.

A single sign-on environment that is configured at the container, organizational unit, or Group Policy level is inherited by all associated directory objects in the hierarchy.

First, enable applications for single sign-on with one user, then copy the applications to the container, OU or Group Policy level for mass deployment. This applies to all SecureLogin configurations, including password policies and preferences. Lower-level settings that you manually configure always override higher-level settings. Therefore, configuration at the user object level overrides all higher level configuration settings. You can manually disable inheritance from objects higher in directory by selecting Yes next to Stop walking here in the Preferences of the desired object. For example you can set this value at the partition boundaries to keep users from needing to walking across WAN links to locate inherited values for SecureLogin.

Distributing Configurations Within Directory Domains

There are two options for distributing the single sign-on-configured environment:

- **Corporate Redirection:** Specifies the object from which the selected object will inherit its SecureLogin configuration settings.
- **Copy SecureLogin Configuration:** Replicates and stores the SecureLogin environment from one directory object to another.

Choose the appropriate option based on the additional information in the following table:

Table 13-1 *SecureLogin Configuration Options*

If	Then
◆ Multiple containers or organizational units require the same SecureLogin environment, and you want to manage configuration from one directory object. ◆ Inheritance, from a higher level object in the directory is not required. ◆ The container or OUs are on the same directory tree. Do not use Corporate redirection across a slower WAN link.	Click **Corporate redirection**.
◆ You want to distribute configurations within the same directory across a slower WAN link. ◆ You want to quickly replicate a complete SecureLogin configuration environment from one object to another in the directory. ◆ You do not want to use XML files to distribute SecureLogin configuration data.	Click **Copy SecureLogin configuration**.

Setting Corporate Redirection

The Corporate Redirection policy distributes SecureLogin settings of a specified object, which can be a container or an organizational unit, to another directory. When this policy is enabled, the recipient directory ignores the SecureLogin settings of its parent directory and inherits the SecureLogin settings of the specified object. The inherited SecureLogin configurations can include enabled applications, password rules, or any other settings.

The Corporate Redirection functionality bypasses the Microsoft Active Directory, NetIQ eDirectory™ inheritance by specifying the source object from which the current object inherits its single sign-on configuration. Although inheritance is redirected to a specific object, such as a container or organizational unit, local user object settings continue to override the inherited settings.

Before you set corporate redirection, the Administrative Management utility must be active.

Corporate redirection cannot be applied to a group object because they are not part of the hierarchy but linked to it.

Consider the following example:

1. Create two directory containers (OU's) under `O=novell`:
   - `ou_apps`
   - `ou_users`
2. Create a user (`user1`) in `ou_users` (`user1.ou_user.novell`).
3. Create SecureLogin applications and, or define settings on the `ou_apps.novell` container.
4. Set corporate redirection on `ou_users.novell` to point to `ou_apps.novell`. The following is seen:
   - `user1` has applications and settings defined at `ou_apps.novell`.
   - `user1` also has its own applications and settings.

You can configure the **Corporate redirection** preference only to be redirected to a specific organizational unit or container.

- When set to a user, the user does not inherit any SecureLogin preferences from their nominal hierarchy but from the other organizational unit or container.
- When applied to an organizational unit or container, any user in that object does not inherit SecureLogin preferences from its container settings. It inherits from the other organizational unit or container.

To get the correct inheritance, users must be granted the correct rights to inherit from other object. The inheritance process stops at the redirected container. There is no inheritance from the redirected object's hierarchy.

In the following example, the Finance organizational unit is redirected to inherit the SecureLogin configuration from the Development organizational unit.

1. Launch the Administrative Management utility (iManager, SLManager, or MMC snap-ins).
2. Click **Advanced Settings**. The Advanced Settings pane is displayed.
3. Specify the full distinguished name of the object in the **Corporate redirection** field.

   NOTE: The full distinguished name is required to uniquely identify the container or organizational unit.

   In this example, the Development organizational unit (ou=development,dc=training7,dc=com)
4. Click **Apply**.
5. Click **OK**.

   Click **Applications** to view the application definitions inherited from the object. Click **Preferences** to view the inherited preferences. In this example, the preferences inherited from the Development ou.

   Ensure that you do not overwrite administrator settings when distributing SecureLogin configuration environments. For example, if you set the preference **Allow users to view and change settings** to **No** and then copy this to the container or organizational unit as part of a SecureLogin environment, including the Administrator user object, the administrator cannot view or change SecureLogin settings because they reside in that organizational unit. To prevent this from happening, all administrator user objects should be located in a separate organizational unit, and administrator preferences should be manually configured.

Setting Corporate Redirection with eDirectory

IMPORTANT: This is required if you wish to use group management after upgrading to SecureLogin 6.1, 6.1 SP1, or later.

To use the eDirectory group membership feature, you must run the new ndsschema tool to correctly set the group, user, and container assignments before upgrading to SecureLogin.

You can resolve this in one of the following ways:

- Run the ndsschema tool to assign the necessary rights and attributes or schema assignments to the group objects.
- Manage through iManager by running the SecureLogin 8.5 plug-in.

Configuring Groups Within eDirectory

With the introduction of the eDirectory group membership feature in the SecureLogin 6.1 release, you must make additional attribute assignments to the group objects. This is primarily required when users are using different administrative management utilities such as NWAdmin, ConsoleOne, or iManager.

1. Launch the Administrative Management utility (iManager, or SLManager).
2. Specify the distinguished name of the container object you want to modify.

   NOTE: SecureLogin only supports configuring group memberships within a container object.

3. Select **Advanced Settings** > **Configured Groups**. The Group Configuration dialog box is displayed.
4. Click **Add**. The Adding a group dialog is displayed.

   The list shows the group objects configured in the current object.
5. Provide the distinguished name of the group object.
6. Click **OK** to add the new group object. The Group Configuration dialog is displayed.

   Use the **Up** and **Down** options to promote or demote the order in which the group policies are applied.

   NOTE: Within the Group Configuration, the higher group takes precedence.

   Configured groups can only be set against containers like O and OU and not set against a user object. In such a case, contrary to the earlier statement, the higher container takes the lower precedence.

   After you have configured single sign-on settings for a Group, the configuration is not reflected in iManager when looking at the SecureLogin information for any of the assigned group members. For example, the group “Everyone” might contain a single application called “innerweb”.

   When looking at the defined SecureLogin applications from within the properties of a group member, you will not see the "innerweb" application listed. However, the configured application will be available in the client when SecureLogin is launched at the workstation.

Copying a Configuration Across Organizational Units

You can copy an object’s SecureLogin configuration to another object from the **Distribution** pane in the Administrative Management utility. This functionality replicates the SecureLogin configuration internally in the same directory tree.

NOTE: In the following example, the Development organizational unit SecureLogin environment is copied to the Finance organizational unit.

1. Launch the Administrative Management utility (iManager, SLManager or MMC).
2. Click **Distribution**. The Distribution pane is displayed.
3. Click **Copy**. The Copy dialog box is displayed.
4. Under **Select SecureLogin Configuration**, select or clear the appropriate check boxes.

Configuration	Function
Applications	Copies, exports, or imports all configured application definitions, as displayed in the Applications pane.
Credentials	Copies, exports, or imports all credentials as displayed in the Logins pane, excluding passwords.
Password Policies	Copies, exports, or imports password policies as displayed in the Password Policies Properties table
Preferences	Copies, exports, or imports all preferences manually set in the Preferences pane.
Active Passphrase Question	Provides users with a selection of passphrase questions. This option copies, exports, or imports only the passphrase question the user has responded to.

5 In the **Destination Object** drop-down list, click the name of the object or type the full distinguished name in the box.

6 Click **Copy**.

If a predefined application or an application definition currently exists in the destination object, a confirmation message appears. It confirms or rejects the overwriting of the imported data.

7 Click **Yes** or **No** as required.

The selected SecureLogin configuration is copied across to the destination user object, organizational unit or container. A confirmation message appears, advising what information has been loaded to the destination object.

8 Click **OK**.

Creating an Active Directory Group Policy

Group Policy Object Support

Prerequisites:

- SecureLogin is installed with support for group policies.
- The Active Directory Users and Computers snap-in or Group Policy Management Console is open

Using Group Policy object support, you can manage SecureLogin users in Active Directory users at the container, OU, and user object levels.

Group Policy object support is useful for organizations with flat directory structures where a more granular approach is required when applying settings, policies, and application definitions for users. For example, applying a group policy for a global marketing group in a worldwide organization. Several group policies can be defined and applied to any user, group, or container at the directory level. These different policies are then applied to a specific user object or container or organizational unit through the inheritance process.

To limit network traffic during the Group Policy object synchronization, SecureLogin leverages an existing Microsoft Windows feature to specify policy settings that are updated when the group policy object changes.

Edit the WinLogon/GPextensions in the Windows Registry, and set the NoGPOListChanges key to 1.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\GPExtensions\{2893059c-1175-11d9-8088-00e018f97d4d
```

For more information on Microsoft Windows Group Policy configuration, see the Microsoft Web site.

For information on the Registry `NoGPOListChanges` setting, see the Microsoft Web site.

Group Policy Management Console Support

In SecureLogin 6.1, you can see the resultant set of single sign-on policy settings that apply to a particular user object when multiple SecureLogin group policies and organizational unit or user object setting are applied through the Microsoft's Group Policy Management Console (GPMC), which now includes support for Resultant Set of Policy (RSOP).

NOTE: The GPMC must be installed on the administrative workstation where you want to see the resultant set of policies.

Resultant Set of Policy Settings

The Resultant Set of Policy (RSOP) is a feature of a group policy that makes the implementation, troubleshooting, and planning of group policies easier and allows you to plan how the group policy changes might affect a targeted user or computer or remotely verify the policies under effect on a specific computer.

When multiple group policy objects are applied to a given user or computer, the policy can often contain conflicting policy settings. For most policy settings, the final value of the setting is set only by the highest precedent Group Policy object that contains that setting.

RSOP assists directory administrators to understand and identify the final set of policies that are applied as well as settings that did not apply as a result of policy inheritance.

In this version of SecureLogin, you can see the final SecureLogin settings that apply to a user when he or she starts SecureLogin. You have the ability to do the following:

- Retrieve the policy applied to the user object in the Microsoft Management Console.
- Retrieve the policy applied to the user object in the SLManager.
- Define from which policy the setting is inherited.

Definition of a Group Policy Object

IMPORTANT: Group policy functionality is enabled only if it was selected during the installation of SecureLogin in Microsoft Active Directory mode. For more information, see the "Installing and Configuring in Active Directory Environment".

For more information about Group Policy Objects (GPOs), go to the Microsoft Web site.

Policy settings are stored in Group Policy Objects (GPOs). Settings for each GPO can be edited using the GPO Editor from within Microsoft's Group Policy Management Console (GPMC).

When an administrator defines a SecureLogin GPO, they can now use the GPMC to add this group policy or edit and configure the SecureLogin settings.

Adding or Editing a Group Policy Object

Policy settings are stored in Group Policy object settings for each Group Policy object and can be edited using the Group Policy object editor from Microsoft (GPMC).

The group policy functionality is enabled during the installation of SecureLogin in Microsoft Active Directory mode. For more information see, "Installing and Configuring in Active Directory Environment"in the *NetIQ SecureLogin Installation Guide*.

When you define a SecureLogin Group Policy Object, administrative users can use the GPMC tool to add this group policy or edit and configure the SecureLogin settings.

Installing the GPMC Plug-In

With the Microsoft's GPMC plug-in, you can manage core aspects of Group Policy object across enterprises.

For Microsoft Vista (or higher) customers, the GPMC snap-in is already integrated in to the operating system.

Existing Windows XP and Server customers can download the `gpmc.msi` installer package at the Microsoft Web site.Installing the Microsoft GPMC plug-in simply involves running the gpmc.msi installer package.

NOTE: After installation, the **Group Policy** tab that previously appeared on the Property pages of sites, domains, and organizational units in the Active Directory plug-in is updated to provide a direct link to GPMC. The functionality that previously existed on the original **Group Policy** tab is no longer available because all functionality for managing a Group Policy is available through the GPMC plug-in.

Managing Group Policy Objects through the GPMC

Use any of the following methods to open the GPMC plug-in directly:

- Click **Start** > **Programs** > **Administrative Tools** > **Active Directory Users and Computers**. The Active Directory Users and Computers page is displayed.
- In the navigation tree, right-click the appropriate organizational unit, then click **Properties**. The selected organizational unit page is displayed.
- Click **Group Policy**, then click **Open**.

- Click **Start** > **Programs** > **Administrative Tools** > **Group Policy Management**.
- Click **Start** > **Run**. The Run page is displayed.

1. At **Open**, type `mmc`.
2. Click **OK**. The Management Console is displayed.
3. Click **File**.
4. Click **Add/Remove Snap-in**. The **Add/Remove** page is displayed.
5. Click **Add**. The Add Standalone Snap-in page is displayed.
6. Select **Group Policy Management** and then, click **Add**.
7. Click **Close**. The Add Standalone Snap-in page is displayed.
8. Click **OK**. The Group Policy Management page is displayed.

NOTE: When you launch the GPMC for the first time, it loads the forest and domain containing the user object logged in to the computer. You can then specify the forest and domain to be displayed.

When you close the GPMC, it automatically saves the last view and returns that view the next a user opens the console.

Retrieving a Policy Applied to the User Object in GPMC

The definition of the Group Policy Objects are defined by the administrator at the directory level, so changes can now be seen immediately at the OU, container or user object level, depending on the level where the group policies have been applied and the SecureLogin preferences applied.

These settings must follow the rules already defined of inheritance and precedence:

- The **Stop walking here** preference
- The **Corporate Redirection** setting
- The Group Policy object settings and their priorities
- The directory hierarchy settings

The precedence rules are respected and follow the rules already defined:

- The deepest object in the tree has the precedence over any other higher-level object
- The group policies have the lower precedence than all OUs and User objects.

As a consequence of all these processes, the administrator can now see the resultant set of the policies in the user object either through MMC interface or administrative management utilities.

The resultant set of policies are displayed in the bottom left hand corner of the SecureLogin Administration Management utility. They show from which Group Policy the current setting has been inherited.

NOTE: The retrieval of all SecureLogin configuration information is subject to both SecureLogin and native Directory access controls. In the unlikely circumstance that the user has rights to read a Group Policy object but the administrator does not, this system displays incorrect effective configuration information. This is because the administrator simply cannot access the same information as the user, and any mechanism for allowing this would introduce a security problem.

In this specific configuration, if SecureLogin has no way to retrieve the exact policy applied to the user object, then a message is displayed indicating that the information displayed does not correspond to the resultant set of policies applied to this user object. The message **RSOP not available** is displayed in the bottom left side of the Administration Management console.

Retrieving a Policy Applied to the User Object in SLManager

Because the definition of the Group Policy objects are performed by you at the directory level, any changes are now seen immediately at the OU, container, or the user object level.

14 Exporting and Importing Configurations

You can use SecureLogin to export or import unencrypted and encrypted XML files. The export functionality of SecureLogin creates an XML file that is external to the directory. You can import the SecureLogin information from this XML file to different directory types, servers, domains, containers, group policies, organizational objects, and user objects.

NOTE: You cannot export through iManager and import using SLManager or vice versa.

In iManager if you try importing a configuration file exported through SLManager, a Java warning message indicating, java.lang.NullPointerException: null is displayed.

Some of the features explained in this section are available only in SLManager. Such features are explicitly indicated. Otherwise, all the features are available in both the administrative management utilities.

Exporting XML Settings

To export XML settings:

1. Log in to iManager.
2. Select **NetIQ SecureLogin > Manage SecureLogin SSO**. The Manage SecureLogin page is displayed.
3. In the object field, specify your object name, then click **OK**.
4. Click **Distribution**. The distribution details are displayed.
5. Click **Save**. The Configuration for Export dialog box is displayed.
6. Under **Select SecureLogin Configuration**, select the configuration(s) you want to export.

Configuration	Function
Application	Exports all configured application definitions as displayed in the **Application** pane.
Credentials	Exports all credentials as displayed in the Logins pane, excluding passwords.
Password Policies	Exports password policies as displayed in the **Password Policies Properties** table.

Configuration	Function
Preferences	Exports preferences manually set in the **Preferences Properties** tables.

7 Click **Export**. The Select the Applications for Backup page is displayed.

8 Select the applications you want to backup.

9 Click **OK**. The Save File As dialog box is displayed.

10 Provide a name to the file, select the file location, and click **Save**.

Importing XML Settings

1 Log in to iManager.

2 Select **NetIQ SecureLogin > Manage SecureLogin SSO**. The Manage SecureLogin page is displayed.

3 In the object field, specify your object name, then click **OK**.

4 Click **Distribution**. The Distribution details are displayed.

5 Click **Load**. The Select SecureLogin Configuration dialog box is displayed.

6 Under **Select SecureLogin Configuration**, select the configuration(s) you want to import.

Configuration	Function
Application	Imports all configured application definitions as displayed in the **Application** pane.
Credentials	Imports all credentials as displayed in the Logins pane, excluding passwords.
Password Policies	Imports password policies as displayed in the **Password Policies Properties** table.
Preferences	Imports preferences manually set in the **Preferences Properties** tables.

7 Browse to and select the exported XML file.

8 Click **Open** to select the file.

The selected SecureLogin configuration is imported into the receiving directory object

If predefined applications and application definitions currently exist in the receiving object, a confirmation message is displayed to confirm or reject overwrite with the imported data.

9 Click **Import** to confirm or click **Cancel** to reject overwriting with the imported data.

A SecureLogin message is displayed to confirm SecureLogin data is loaded.

Exporting Single Sign-On Data in Encrypted XML Files

Using SecureLogin Manager (SLManager) you can encrypt and password-protect or digitally sign the exported files to ensure the information is secure. Alternatively, an unencrypted file can also be created for unrestricted distribution.

This option is available only through SecureLogin Manager (SLManager).

1. Launch SecureLogin Manager.
2. In the object field, specify your object name, then click **OK**.
3. Click **Distribution**. The Distribution details are displayed.
4. Click Save. The save dialog box is displayed.
5. Select the appropriate options. The following table describes the options:

Configuration	Function
Application	Exports all configured application definitions as displayed in the **Application** pane.
Credentials	Exports all credentials as displayed in the Logins pane, including passwords.
Password Policies	Exports password policies as displayed in the **Password Policies Properties** table.
Preferences	Exports preferences manually set in the **Preferences Properties** tables.
Passphrase Question	Provides users with a selection of passphrase questions. This option copies, exports, and imports only those passphrase questions to which the user has responded.

6. From **Select File Protection**, select **Password protected and encrypted**.
7. Specify the password in the **Password** field.
8. Re-specify the password in the **Verify** field.
9. Click **OK**. The select application to export dialog box is displayed.
10. Select the applications to be exported, then click **OK**.
11. Select a location to save the file.
12. Specify a name for the file.
13. Click **Save**. The selected SecureLogin configuration is saved and a confirmation message appears indicating the information that is saved.
14. Click **OK**.

Importing Single Sign-On Data in Encrypted XML Files

1. Launch SecureLogin Manager.
2. In the object field, specify your object name, then click **OK**.

3 Click **Distribution**. The Distribution details are displayed.

4 Click **Load**. The load dialog box appears.

5 Select the required options. The following table helps you choose.

Configuration	Function
Application	Imports all configured application definitions as displayed in the **Application** pane.
Credentials	Imports all credentials as displayed in the Logins pane, including passwords.
Password Policies	Imports password policies as displayed in the **Password Policies Properties** table.
Preferences	Imports preferences manually set in the **Preferences Properties** tables.
Passphrase Question	Provides users with a selection of passphrase questions. This option copies, exports, and imports only those passphrase questions to which the user has responded.

6 Click **OK**. The open dialog box is displayed.

7 Select the exported encrypted file.

8 Click **Open**. The password dialog box is displayed.

9 Specify the password, then click **OK**.

If a predefined application or an application definition currently exists in the destination object, you get a confirmation message appears for the applications.

Click Yes if you are sure that the imported application definition is preferred over the application definition currently stored on the destination object.

Click No to retain the application definition currently stored on the destination object.

10 If you click Yes, the configuration is copied across to the user object, organizational unit, or container. A confirmation message appears indicating that the information is copied to the destination object.

11 Click **OK**.

Creating a Signing Key for Secure Distribution

After you have configured and tested SecureLogin in an user environment, you can create a digital signing key that is embedded in the .exe file. You can distribute the file through a Web download or e-mail to the users. When users receive the file, they need to double-click the file to load to the local workstation. This updates the following:

- Preferences
- Application definitions
- Password rules
- Credentials

This is collectively known as the SecureLogin configured user environment and, is particularly designed for users who use SecureLogin in standalone mode (such as mobile users) and those who do not frequently connect to the corporate network.

When a digital signing key is created, the key pair is randomly generated by the SecureLogin to increase security.

To create a digital signing key:

IMPORTANT: This feature is available only through SecureLogin Manager.

---

1. Launch SecureLogin Manager.
2. In the object field, specify your object name, then click **OK**.
3. Click **Distribution**. The Distribution details are displayed.
4. Click **Save**. The save dialog box is displayed.
5. Select the required options.
6. Under **Select File Protection**, select **Digitally signed and encrypted**.
7. (Optional) Select Administrative data will overwrite user's data without notification.

   If this option is selected, the users are prompted before overwriting any data with the configuration settings saved in the .msi file.

   IMPORTANT: Selecting this option results in the user data being overwritten with the configuration setting in the `.msi` file for any items that are present in both the user's local configuration and the administrative configuration (`.exe` file).

   For example, if a user have an application definition configured locally, and a predefined application definition is supplied in the `.exe` file, the `.exe` file application definition overwrites the user's application definition without notification.

   However, for example, if a user has configured a Hotmail application definition locally, and a predefined application is not supplied in the `.msi` file, the user's Hotmail application definition is not changed.

   ---
8. Click **Manage Keys**. The Manage signing keys for secure file distribution dialog box is displayed.
9. Specify a name for the key in the **Generate Digital Signing Key** field.
10. Click **Create**.
11. From the **Key List**, select the newly created key.
12. Under **Install**, click **Install Package**. The Load Settings dialog box is displayed.
13. Browse to locate the distribution file (`.msi` file) in which you want to embed the key.
14. Click **Open**. A confirmation message that the key is embedded in the `.msi` file is displayed.
15. Click **OK**.

    You can now distribute and install the `.msi` file on the user's machine. This allows them to import signs that are signed and encrypted.

    After the keys are created, they must not be deleted because they are randomly generated. They key used must correspond to the key that is been previously packaged and with the distributed installer.

Locally Installing a Digital Signing Key

The **Manage signing keys for secure file distribution** dialog box provides a tool to install a digital signing key locally, enabling loading of XML files generated using this key

1. Log in to iManager.
2. Select **NetIQ SecureLogin** > **Manage SecureLogin SSO**. The Manage SecureLogin page is displayed.
3. In the object field, specify your object name, then click **OK**.
4. Click **Distribution**. The Distribution details are displayed.
5. Click **Manage Keys**. The Manage signing keys for secure file distribution dialog box is displayed.
6. Specify a name in the **Generate Digital Signing Key** field.
7. Click **Create**.
8. From the **Key List**, select a new key.
9. Under **Install**, select **Install Locally**. A confirmation message appears.
10. Click **OK**.

15 Using The slAP Tool

This section provides information on the following:

About The slAP Tool

The SecureLogin Attribute Provisioning (slAP) tool uses command line options to allow SecureLogin to leverage user data from an organization’s provisioning system. Using the slAP tool, you can import data, in XML format from third-party applications into the SecureLogin user’s datastore as well as export information (except passphrase answers).

Data that can be manipulated includes:

- User variables
- Application definitions
- Organizational settings
- Password
- Password policies
- Logins
- Passphrase questions

The slAP tool command operates as a provisioning tool between SecureLogin data in a directory and in an XML file. The XML schema used is the same as the Copy Settings GUI importer/exporter. In addition to copying settings, the slAP tool can extract usernames.

For example, an organization with 10,000 users in a SAP system, implementing SecureLogin can speed deployment significantly by automating the initial user login. To do this, use a file containing multiple users’ username and password combinations from SAP, and use the slAP tool to import the file into the SecureLogin datastore as a bulk process. The slAP tool removes the requirement for each user to enter credentials on the first log in to SecureLogin.

If the slAP tool is used to import data into SecureLogin from either an encrypted or an unencrypted file, and any preferences are set that require the SecureLogin version 6 data store format, then the datastore version must be specified in the file. Preferences that require the version 6 format are:

- EncryptionType
- NRKeySource
- StoreDataOnSmartCard
- UseEnhancedProtectionByDefault

The datastore version is set as:

```
<preference>
   <name>AppliedSSODataStoreVersion</name>
   <value>600000</value>
   <isdatastore/>
</preference>
```

If the value of this preference is not set to 6, 6.0 or 600000 then an error message is returned from the slAP tool: *Cannot import version 6 datastore preferences into a lower versioned datastore.*

When the slAP tool is used for initial provisioning of SecureLogin user accounts, before any SecureLogin data has been stored for users, the XML file must include a passphrase question and response. This question/response can be the same for each user and can be changed by the user after deployment.

NOTE: SecureLogin does not need to be running to use the slAP tool.

The slAP Syntax

```
slaptool [-hlaspcPef] -r object_name_file | -o "object" [file ...]
```

The following table describes the command options.

Command	Description
`-h`	Displays a help message and exits (all other options are ignored).
`-l`	Excludes user IDs.
`-v`	Excludes variables.
`-a`	Excludes applications.
`-s`	Excludes settings.
`-S`	Include passwords. Only applies to export operations which include symbols and must be used in conjunction with `-E` for security.
`-p`	Excludes password policies.
`-E password`	Specifies that the generated XML should be encrypted or decrypted using the supplied password. Password must be at least 8 characters long.
-f	Use the current user, for export allow inclusion of password credentials. (cannot be used with `-r` or `-o`).
`-c`	Excludes credsets.

Command	Description

`-d`	Performs delete rather than import.
	
	For example:
	
	◆ To delete logins:
	
	`slaptool -d -o "cn=abc,dc=123" -l`
	
	◆ To delete applications:
	
	`slaptool -d -o "cn=abc,dc=123" -a`
	
	◆ To delete password policy:
	
	`slaptool -d -o "cn=abc,dc=123" -p`
	
	NOTE: You can also use `-f`, to perform delete operation for the current user.
	
	For example, `slaptool -d -f -l`.

`-e`	Performs an export rather than an import.
`-r`	object_name_file Specifies a file containing line-delimited object names on which to perform the operation.
`-o`	object Specifies a particular object on which to operate.

[file]	Specifies one or more `.XML` files from which to read data (or to write to for exporting). No file specification. It reads and writes data from and to the stdin and stdout.
	
	For example:
	
	`./slaptool.exe -o "cn=bernie, cn=netiq, dc=testdomain, dc=com" initial_setup.xml`
	
	This reads userIDs, applications, settings and password policies from the file initial_setup.xml and writes them out to the object:
	
	`"cn=bernie, cn=netiq, dc=testdomain, dc=com"`

`-P`	Exclude Passphrase.
`-k` [password]	Enables the creation of a passphrase answer for individual users in LDAP and Microsoft Active Directory environments. It is mandatory for users to save a passphrase answer on first log in to SecureLogin. The slAP tool requires password authorization to save user data. The `-k` switch provides the user password, enabling automated creation of the passphrase answer. This answer can be manually changed by users after provisioning. For example, the following command is used to import user data and a passphrase question and answer combination: `slaptool.exe -k password -o context filename.xml` This reads userIDs, applications, settings, and password policies from the file `initial_setup.xml` file and writes them out to the object: `"cn=writer, cn=netiq, dc=testdomain, dc=com”`

NOTE

- If the `-P` switch was not used during the export operation, then you must use either `-P` or `-k` switch during the import operation.
- If `-P` switch was used during export operation, then `-P` or `-k` switch is not required during the import operation.
- When using the slAP tool in an eDirectory mode and when Novell Client is installed, use the following syntax:

  ```
  slaptool <command option> -o <user DN in NDS format> -a
  ```

  For example:

  ```
  slaptool -d -o "abc.mytestou.novell" -a
  ```

slAP Tool Example

The following Perl application definition, created for the example organization discussed previously, assumes that usernames and passwords are stored in a text file named `listofnames.txt`. There is one space between each username and password pair per line.

A XML file, such as the “XML File Example” on page 117 is required to run this application definition, containing the data for import. Where the data is customized on a per user name basis, the string to be substituted is replaced with *usernamegoeshere*.

For example:

```
************************************************
open FILE,"listofnames.txt";
foreach (<FILE>) {
chomp;                  # Clean string
@lines = split(/\n/);   # Split up string
for each $l (@lines) {
    @fields = split(/\s/);
    $name = $fields[0];
    $pass = $fields[1];
    open DATAFILE,"source.xml";
    open OUTFILE,">data.xml";
    foreach (<DATAFILE>) { # Write up a file specific to this user
        s/\*usernamegoeshere\*/$name/;
        s/\*passwordgoeshere\*/$pass/;
        # Any other variable substitution can be done here too...
        print OUTFILE "$_";
   }
    close DATAFILE;
    close OUTFILE;
   system "slaptool.exe -k \"$pass\" -o
\"CN=$name.O=myorg.T=OURCOMPANY\" data.xml";
        }
}
close FILE;
unlink 'data.xml';
************************************************
```

Using an XML file called `source.xml`, run the application definition with the data that is to be imported. For example, you can manually export data from a single user setup with the value for the username replaced with the string "*usernamegoeshere*".

NOTE: The example application definition does not include error handling.

XML File Example

```
<?xml version="1.0"?>
<SecureLogin>
   <passphrasequestions>
      <question>Please enter a passphrase for SLAP testing.</question>
   </passphrasequestions>
   <passphrase>
      <activequestion>Please enter a passphrase for SLAP
testing.</activequestion>
      <answer>passphrase</answer>
   </passphrase>
   <logins>
      <login>
         <name>fnord</name>
         <symbol>
            <name>username</name>
            <value>bob</value>
         </symbol>
         <symbol>
            <name>Password</name>
            <value>test</value>
         </symbol>
      </login>
<login>
         <name>notepad.exe</name>
         <symbol>
            <name>username</name>
            <value>asdf</value>
         </symbol>
         <symbol>
            <name>Password</name>
            <value>test</value>
         </symbol>
      </login>
      <login>
         <name>testlogin</name>
         <symbol>
            <name>username</name>
            <value>Novell</value>
         </symbol>
         <symbol>
            <name>Password</name>
            <value>test</value>
         </symbol>
      </login>
    </logins>
</SecureLogin>
```

16 Using The slMigrationHelper Tool for Datastore Migration

This release of NSL introduces the `slMigrationHelper` tool to migrate data from an existing datastore to a new datastore. This tool is an enhanced version of the existing SLAP tool and helps in seamless migration with minimal disruption of service.

The data migrated during the migration process includes:

- User variables
- Application definitions
- Organizational settings
- Password policies
- Logins
- Passphrase questions and answers

The benefits are:

- You do not have to install NSL in a new mode. You only need to export the data from an existing installation and then import the data to configure NSL with the new datastore.
- Single restart – With this tool, you restart your workstation only once to complete the datastore migration. This is an improvement over the earlier migration process where multiple workstation restarts are required.
- Seamless Migration – The entire process of exporting the data, reconfiguring NSL to the new datastore, and importing data is seamless and requires minimal manual intervention.
- No uninstallation required during migration – You can continue the migration process without having to uninstall the existing NSL installation. This causes minimal disruption of service.

Supported Datastores

Datastore migration is supported for the following directories :

- Active Directory – Windows 2003/Windows 2008/Windows 2012
- eDirectory 887
- OpenLDAP
- ADAM

Process of Datastore Migration

1. Access the tool from the `SecureLogin\Tools\Administration\Provision Tools` folder.
2. Run the tool with the option to export all the data and specify the new datastore option. The data is exported to an XML file.

3. Run the tool again with the option to import all the data.

   Restart NSL to load it in the new datastore mode.

The valid options for the `slMigrationHelper` tool are:

Option	Description
-m	Use this option to modify an existing NSL installation. This option is used in combination with option `-u` to specify the new datastore. For example: If you are already on eDirectory, run the slMigrationHelper tool with option `-m` to specify that the existing installation has to be modified to run with a new datastore. `slmigrationhelper.exe -m [`*`Datastore options`*`]` You can also specify additional options during the modify process. For example: `slmigrationhelper.exe -m ADDLOCAL=SeamlessLDAPGina`

-u <path to the installation program>	Use this option to upgrade NSL from an old version to a newer version and change the datastore.
	
	To upgrade and change the datastore, use this option in combination with option `-t` to specify the new datastore.
	
	For example:
	
	`slmigrationhelper.exe – u <path to the 8.0 installer file> - t [datastore]`
	
	You can also choose to upgrade to a newer version without changing the datastore.
	
	For example:
	
	`slmigrationhelper.exe – u <path to the 8.0 installer file>`
	
	You can also specify additional option during the upgrade or modify process.
	
	For example:
	
	`slmigrationhelper.exe -u<path to the 8.0 installer file> ADDLOCAL=SeamlessLDAPGina`

Option	Description

-t <datastore>	Use this option to specify the datastore you want to migrate to.
	
	This option is used in combination with option `-m`.
	
	The valid datastore's are:
	
	◆ MAD
	◆ LDAP
	◆ LDAPSecretstore
	◆ NDS
	◆ Secretstore
	
	For example:
	
	To switch to LDAP mode and install in GINA/Credential Provider mode, the command is:
	
	`slmigrationhelper.exe -u C:\NetIQSecureLogin.exe -t LDAP ADDLOCAL=SeamlessLDAPGina`
	
	To switch to LDAP mode and install in Credential Manager mode, the command is:
	
	`slmigrationhelper.exe -u C:\NetIQSecureLogin.exe -t LDAP -q ADDLOCAL=SeamlessLDAPCred`
	
	To switch to LDAP mode and also specify an LDAP server address, the command is:
	
	`slmigrationhelper.exe -m -t LDAP LDAPSERVERADDRESS=`*`127.0.0.1`*
	
	You can also specify features to be installed using the APPENDLOCAL property.
	
	`slmigrationhelper.exe -m -t LDAP LDAPSERVERADDRESS=`*`127.0.0.1`*
	`APPENDLOCAL=DAS`
	
	IMPORTANT: Installing in any LDAP or LDAPv3 mode requires NICI to be installed.
	
	If you are modifying the datastore from an existing one to LDAP and NICI is not installed on your workstation, use the -u option to specify the path to SecureLogin installer.
	
	For example:
	
	`slmigrationhelper.exe -u `*`C:\NetIQSecureLogin.exe`*` -t `*`LDAP`*` -q`
	
	This switches the datastore to LDAP and installs NICI in the quiet mode.

Option	Description
-f <path for the XML file>	Use this option to specify path to the file that will contain all the exported data. All the data is stored in an XML format. For example: `slmigrationhelper.exe – f <path to the file>`
-i <path to the XML file>	Use this option to import previously exported data. For example: `slmigrationhelper.exe – i`
-E	Use this option to encrypt the exported data. If you do not specify a password, the default password `changeit` is used for encryption. For example: `slmigrationhelper.exe – E <password>`
-P	Use this option to exclude import/export of passphrase information. If you have excluded passphrase import and export, during installation the user has to configure the passphrase information. For example: `slmigrationhelper.exe -m -f <path to the XML file> -P`
-r	Use this option to invoke importing of user data from the XML file. When you use this option, HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce is updated. For example: `slmigrationhelper.exe -r`
-q	Use this option to run the installer in quiet mode. By default all the installation program screens are displayed during the migration process. For example: `slmigrationhelper.exe -q`
-h	Use this option to display help for the `slmigrationhelper` tool.

17 Managing the Workstation Cache

This section provides information on the following:

About the Workstation Cache

The SecureLogin cache is an encrypted local copy of SecureLogin data. It allows users who are not connected to the network (or working offline using a laptop) to continue to use SecureLogin even if the directory becomes unavailable.

User data includes credentials, preferences, policies, and SecureLogin application definitions, except when you use a smart card for storing credentials. By default, a cache file is created on the workstation as part of SecureLogin installation. The cache file stores user data locally and is synchronized regularly with the user’s data in the directory. You can set the synchronization period in the Administrative Management utility. You can also disable the offline cache, forcing all SecureLogin data to be stored in the directory.

Depending on the type of installation, the cache is stored:

- In the users profile directory. For example

  `%APPDATA%\SecureLogin\Cache`

  On Microsoft Windows Vista and higher: `%APPDATA%` represents

  `C:\Users\<Username>\Appdata\Roaming\`

  On XP systems: `%APPDATA` represents

  `C:\Documents and Settings\<Username>\Application Data\`

  or
- In the `%ProgramFiles%`. For example:

  `C:\Program Files\NetIQ\SecureLogin\Cache`

Directory and workstation caches are synchronized regularly, by default every five minutes, and whenever the user logs off or on to the workstation. When changes are made, either by the user on the workstation or the administrator in the directory, single sign-on user data is compared and updated during synchronization. Any settings configured by the user through the Credentials Management tool on the local workstation take precedence over those made in the directory.

If you require full administrative control of a user’s SecureLogin environment, you can disable the user's access to administration tools through the settings in the Preferences Properties table. This prohibits users from overriding your changes while configuring changes on the workstation.

NOTE: The SecureLogin cache refresh interval is by default five minutes. You can change the default in the Preferences Properties table.

Because SecureLogin data is stored in the directory, existing directory backups also back up SecureLogin data. In addition, the local cache synchronizes with the directory for further redundancy of data. Backing up or restoring by using the SecureLogin menu options is typically performed by users who have been disconnected from the network for long periods of time, such as weeks or months.

Using workstation backup and restore, users can securely back up their SecureLogin cache in stand-alone or directory deployments. All user data, including passwords and passphrases, is saved in a password-protected, encrypted XML file.

Creating a Backup File

1. In the notification area (system tray), right-click the NetIQ SecureLogin icon, then select **Advanced > Backup User Information**. The Save Settings dialog box is displayed.
2. Select a folder to store the backup file. The file can be stored in any location.
3. In the **File name** field, specify a name for the backup file.
4. Click **Save.** The Password dialog box is displayed.
5. In the **Password** field, specify a password.

   SecureLogin verifies if the password entered is at least 8 characters long. If the password does not meet this requirement then the user is prompted to re-enter the password with the required length.
6. Click **OK**.

   The encrypted and password-protected backup file is saved, and a confirmation message appears.
7. Click **OK**.

Deleting the Workstation Cache

IMPORTANT: Before restoring the backup file, you must delete the cache file on the workstation. In directory environments, you must also delete the user object data in the directory.

1. Right-click the Windows **Start** button, then click **Explore**.

   Ensure that you have selected `Show hidden files and folders` in the Windows Folder Options dialog box.
2. Depending on the type of installation, browse to `C:\Program Files\SecureLogin\Cache`

   or

   `%APPDATA%\SecureLogin\Cache`

   On Microsoft Windows Vista, browse to `C:\Users\<Username>\Appdata\Roaming\SecureLogin\Cache`
3. Delete the cache directory.
4. Close Windows Explorer.

Restoring the Local Cache Backup File

1. In the notification area (system tray), right-click the SecureLogin icon, then select **Advanced > Restore User Information**. The Load Settings dialog box is displayed.
2. Select the backup file.
3. Click **Open**. The Password dialog box is displayed.
4. In the **Password** field, specify the password.

   If the specified password is correct, SecureLogin processes the file to restore the user's data. If one or more application are already defined, a series of messages asking the user whether to overwrite the existing workstation file, appears.
5. Select **Yes** to overwrite the file and continue restoring the local cache backup file.
6. After the completion of the restoration process, a confirmation message appears confirming that the cache is successfully loaded to the local workstation cache.

NOTE: If password policies already exist, ignore the wrong error message `0 password policy` that is shown when restoring user data.

---

18 Auditing

This section contains the following information:

About Auditing Tools

SecureLogin provides monitoring functionality with Simple Network Management Protocol (SNMP) trapping and Windows event logging. SecureLogin’s support for both of these auditing tools allows you to choose a preferred auditing application and to integrate event monitoring into your current SNMP functionality. Event alerts are activated through SecureLogin application definitions. An understanding of application definition is useful to enable event monitoring.

About SNMP Auditing

To understand how to use SecureLogin with SNMP, see **SecureLogin Guidelines for SNMP Trapping** under Additional Resources (https://www.netiq.com/documentation/securelogin-86/#addres) at SecureLogin Documentation (https://www.netiq.com/documentation/securelogin-86/).

About Windows Event Log Alerts

Windows event log alerts are activated by following the same procedure as SNMP alerts. The `Logevent.exe` application is activated through the `Run` command in an application definition.

Windows event logging from SecureLogin requires that the Windows Event Log system is active on the computer receiving the alerts, along with the executable `Logevent.exe` on each audited client workstation, to generate the alerts.

NOTE: `Logevent.exe` is included in the Windows 2000 Resource Kit. Microsoft licensing regulations apply.

For details, visit the Microsoft Support Web site.

Creating a Windows Event Log Alert

The following procedure uses the Windows Notepad application as an example.

1. In the notification area (system tray), double-click to open the SecureLogin Client Utility.
2. Click **Applications**.

3 In the right pane, double-click the application description (in this example, Untitled-Notepad). The Application Pane is displayed.

4 Click the **Definition** tab. The application definition editor is displayed.

5 The command syntax to execute `LogEvent.exe` is:

```
logevent -m \\computername-s severity-c categorynumber-r source-e eventID-
timeout"event text"
```

For definitions of the command parameters and see, Microsoft Support Website

6 After `EndDialog`, specify the `LogEvent` command for the required alert.

For example:

```
Run "C:\Program Files\Resource Kit\LogEvent.exe –m SecureLogin –s –e 99"Notepad
has started”
```

This command requests an alert to be sent to the console with a security level of W – warning and event ID number 99.

7 Click **OK**.

8 Start Notepad. The alert is sent to the Windows Event Log system.

19 Audit Configuration for Sentinel

SecureLogin integrates with NetIQ Sentinel for auditing. The events are logged to Windows Event Log from which the auditing server such as NetIQ Sentinel can fetch the event logs. In the previous releases the connection to the auditing server was outbound. That is, the Platform Agents sent the event logs to the Novell Audit.

In SecureLogin, the connection to the auditing server is inbound. The events are logged to Windows Events Log from where the auditing server fetches the event logs. In the following sections, we describe the configuration to enable auditing through Sentinel.

This section consists of:

Windows Event Log: An Overview

Windows event logging is a system service used by the Windows operating system to record the occurrence of system events. Events range from resource tracking of failing device drivers to security-related actions such as attempts to access files, directories, printers, or other system objects that are under audit control. The Windows security event log monitors events generated by system security and auditing processes.

By default, **Windows Security Event Auditing** is turned off.

The Windows Event Viewer is the primary tool for viewing the event logs found on Windows systems.

WMS Connector

The Windows Monitoring Service (WMS) connector facilitates integration between Sentinel Collectors with Microsoft Windows event sources. For SecureLogin, we use the SecureLogin Collector. The collector is available at Sentinel Connector and Collector Web site. (http://support.novell.com/products/sentinel/secure/sentinelplugins.html)

For detailed description on connectors, refer Understanding Event Source Management (http://www.netiq.com/documentation/sentinel61/s61_user/?page=/documentation/sentinel61/s61_user/data/) in Sentinel User Guide. (http://www.netiq.com/documentation/sentinel61/s61_user/?page=/documentation/sentinel61/s61_user/data/)

Configuring Auditing

The configuration of auditing, with the SecureLogin Collector, differs for workstations in Active Directory environments and non-Active Directory environments. The configuration involves enabling audit for the target system and configuration of the appropriate accounts to access the Windows Event Logs remotely by Sentinel. The following are the high level configuration procedures for both scenarios:

- “Monitoring a System in a Domain Environment” on page 130
- “Monitoring a System in a Non-Domain Environment” on page 132

For detailed information, see the WMS Connector document at the Sentinel Connector and Collector Web site. (http://support.novell.com/products/sentinel/sentinel61.html)

Monitoring a System in a Domain Environment

In a domain environment, a domain account must be created that has the policy rights to access the Windows Security Event logs on the remote Event Sources. This domain user account must be recognized by the Event Sources either as a user within the domain, or a user within one of the groups referenced on the server.

- “Configuring Events Logged by Windows Event Log” on page 130
- “Configuring Users to Collect Windows Event Log Remotely” on page 130
- “Setting up the Windows Management Instrumentation Service” on page 131
- “Configuring Domain Account User COM/DCOM” on page 131

Configuring Events Logged by Windows Event Log

Use the following procedure to enable basic Windows event logging for use with Windows Collectors. To collect data from a different application that writes to the Windows Event Log, refer to the documentation for the associated Collector. For details, see the Sentinel Connector and Collector Web site. (http://support.novell.com/products/sentinel/sentinel61.html)

To configure the Sensor to report Events to Security Log:

1. Log on to Windows with an account that has Administrative rights.
2. Click **Start** > **Settings** > **Control Panel**.
3. In Control Panel window, double-click **Administrative Tools**.
4. Double-click **Local Security Policy**; expand **Local Policies**, then double-click **Audit Policy**. A list of policies displays.
5. Double-click a specific audit policy to edit the security settings.
6. In Local Security Setting window, select **Success/Failure** check boxes.
7. Click **OK**.

Configuring Users to Collect Windows Event Log Remotely

1. From the **Event Source**, click **Start** > **Settings** > **Control Panel**.
2. In the Control Panel window, select **Administrative Tools** > **Local Security Policy** > **Local Policies** > **User Rights Assignment** > **Manage auditing and security log**.
3. Click **Add**.

4 From the **Select Users/Groups** window, click the **Look in field**, then select the domain with the account to be used for collecting the security event log information.

5 Double-click the account to be used, then click **OK**.

6 In the Local Security Policy Settings window, click **OK**.

The new policy setting takes effect after you restart the system.

NOTE: If domain-level policy settings are defined, they override local policy settings.

Setting up the Windows Management Instrumentation Service

1 Log on to the remote computer; from the Task bar, click **Start** > **Settings** > **Control Panel**.

2 In the Control Panel window, double-click **Administrative Tools** > **Computer Management**.

3 In the Computer Management window, on the **Tree** tab expand **Services and Applications**; right-click **WMI Control**, then select **Properties**.

4 In WMI Control Properties window, select the **Security** tab.

5 Select the **Root** folder, then click **Security** to open the Security for Root dialog.

If the User or Group that needs the remote WMI access does not appear in the list, click **Add**.

6 From the Select Users, Computers, or Groups window, select the user or group that needs remote WMI access, then click **Add**.

7 After you finish selecting users or groups, click **OK**.

8 Select the newly added user or group and ensure that they have at least the following permissions depending on what type of Event log you want to access:

- Execute Methods
- Provider Write
- Enable Account
- Remote Enable

9 With the user or group still highlighted, click **Advanced** to open the Access Control Settings for Root window.

10 Select the group, then click **View/Edit**, to open the Permission Entry for Root dialog.

11 From the **Apply onto** list, select **This namespace and sub namespaces**.

12 Click **OK** on each dialog until you return to the Computer Management window.

13 Restart the WMI service. For more information on starting the WMI service refer Starting and Stopping the WMI Service (http://msdn.microsoft.com/en-us/library/aa826517(v=vs.85).aspx)

Configuring Domain Account User COM/DCOM

The procedure to configure domain account user COM/DCOM differs from based on the platform on the SecureLogin workstation. Refer the WMS Connector document at the Sentinel Connector and Collector Web site. (http://support.novell.com/products/sentinel/sentinel61.html) for detailed configuration information.

Monitoring a System in a Non-Domain Environment

In a non-domain environment, local accounts must be created on both the Collector Manager system and on the Event Source. These accounts must have the same username and password.

- “Configuring Events Logged by Windows Event Log” on page 132
- “Configuring Users to Collect Windows Event Log Remotely” on page 132
- “Setting up the Windows Management Instrumentation Service” on page 132
- “Configuring Domain Account User COM/DCOM” on page 132

Configuring Events Logged by Windows Event Log

Refer “Configuring Events Logged by Windows Event Log” on page 130 in “Monitoring a System in a Domain Environment” on page 130.

Configuring Users to Collect Windows Event Log Remotely

In a non-Active Directory environment you must create a user account on each event source, that is, each workstation running SecureLogin. This same username and password must also be configured on the Collector Manager machine.

On Collector Manager machine this user must be part of Administrator group.

Refer “Configuring Users to Collect Windows Event Log Remotely” on page 130 in “Monitoring a System in a Domain Environment” on page 130.

Setting up the Windows Management Instrumentation Service

Refer “Setting up the Windows Management Instrumentation Service” on page 131 in “Monitoring a System in a Domain Environment” on page 130.

Configuring Domain Account User COM/DCOM

Refer “Configuring Domain Account User COM/DCOM” on page 131 in “Monitoring a System in a Domain Environment” on page 130.

Logging Events from LDAP

You must configure the registry to enable logging from LDAP.

To log events from SecureLogin LDAP authentication module:

1. Create a registry value at `HKEY_LOCAL_MACHINE\Software\Novell\Login\Ldap`

   The following events are logged:

   ```
   Event ID 1 Informational: NSL user login

   Event ID 2 Informational: LDAP user password change

   Event ID 3 Warning: Workstation unlocked by different User
   ```

2. Open the nslevtsvc.ini file in a text editor, and add the following section about the new information source:

```
[wmi]
event_source=LDAPAuth
```

The following events are logged:

```
Event ID 1 Informational: NSL user login
Event ID 2 Informational: LDAP user password change
Event ID 3 Warning: Workstation unlocked by different User
```

20 Administering Desktop Automation Services

The module `ARS.exe` is the primary component of Desktop Automation Services. Configuration of the service is performed through the use of an XML file. The XML document can be obtained either locally on the workstation or through directory services. The XML document is called the action file and the file is named `actions.xml`.

See an example file in “Example of an Action File” on page 158

Each action is a set of configurable user-level operations such as mapping a drive, testing for an authenticated connection to a directory, and even running/ shutting down an application. The flexibility of the code to test for conditions and have the action triggers such as hot keys, provide tremendous flexibility to change the behavior of the workstation to fit your needs.

The `ARSControl.exe` runs as a Windows service. The `ARSControl.exe` then parses the `actions.xml` file and stores the configuration in memory. All actions performed by `ARS.exe` and `ARSControl.exe` are recorded in a `DASlog.txt` log file at different configurable levels of details.

After you have configured the `ARS.exe` application, it’s actions are available individually or in combination with the scripting interface that is available on Windows. For example, VBScript, JavaScript, login scripts, SecureLogin scripts and batch files.

Actions and Descriptions

Each instance of Desktop Automation Services is configured by an XML document that defines events and desired actions.

The following table describes the elements that might be used to compose a Desktop Automation Services XML input document.

Unless otherwise specified, all XML attributes listed for a given element are required for that element.

Table 20-1 *Desktop Automation Services XML Description*

XML Tags	Description

`application-runner-script`	This is the parent element for an Desktop Automation Services input document.
	
	`application-runner-script` has no attributes.
	
	`application-runner-script` can contain any number of action elements.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<run-application application= "notepad.exe" interval="500" serial="true" parameters="" />`
	`</action>`
	`<action-triggers>`
	`<on-inactivity-timer interval1="100" messagebox="Your " interval2="8" action-name="sample-action"/>`
	`</action-triggers>`
	`</application-runner-script>`

`action-triggers`	This element is a parent (container) for `action-trigger` elements such as `on-nds-login`, or `on-hot-key`.
	
	`action-triggers` enables Desktop Automation Services executables to respond to workstation events by triggering specified actions as defined in the input document.
	
	`action-triggers` has no attributes.
	
	`action-triggers` can contain any of the following child elements:
	
	◆ on-inactivity-timer
	◆ on-nds-login
	◆ on-ldap-login
	◆ on-hot-key
	◆ on-screen-saver
	◆ on-cardmon
	◆ on-Tap-cardmon
	◆ on-ad-login
	
	For Example:
	
	`<action-triggers>`
	`<on-nds-login action-name="LoginInAction" tree="NCCD_TREE_1"/>`
	`</action-triggers>`

XML Tags	Description
on-inactivity-timer	on-inactivity-timer has four attributes: ◆ interval1: is the time of executing an action ◆ interval2: is the time after which warning dialog will be displayed. ◆ messagebox: contains the message to be displayed in the warning. This command element provides information to Desktop Automation Services on the action to be performed if the workstation is inactive for more than the specified period of time. At the end of the countdown period, a specified action such as Close all programs or Lock the Workstation can be invoked. If a mouse or keyboard action is detected, the countdown timer stops and resets until the next inactivity is detected. For Example: <?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <run-application application="notepad.exe" interval="500" serial="true" parameters="" /> </action> <action-triggers> <on-inactivity-timer interval1="100" message box="Your" interval2="8" action-name="sample-action"/> </action-triggers> </application-runner-script> ◆ interval1 is the time of executing an action. ◆ interval2 is the time after which warning dialog will be displayed. ◆ messagebox contains the message to be displayed in the warning. ◆ action-name is the name of the action to be executed. Specify the inactivity timer in seconds. For example, 10 seconds. **NOTE:** You must specify only numbers for interval values in the syntax. If you use special characters in the action.xml file, it does not behave as expected. The on-inactivity-timer is implemented to work with positive numbers. If a negative number or special character is specified, it will behave erroneously. The on-inactivity-timer functions only if the network login is present.

XML Tags	Description

`on-nds-login`	This element defines an action trigger that is activated when a user logs in to eDirectory through the Novell Client GINA/Credential Provider. If a user logs in to the tree, an action trigger invokes Desktop Automation Services. It tests the primary connection to see if the current tree matches the configuration. If it matches, Desktop Automation Services executes the configured action identified by the action name attribute value.
	
	`on-nds-login` element must be contained within an `action-triggers` parent element.
	
	`on-nds-login` has two attributes:
	
	◆ **action-name:** The name of an action defined in the input document that is executed when a user logs in to the tree named by the tree attribute. The action-name must be contained in double quotes.
	◆ **tree:** Connections are tested periodically to see if they are linked to the tree named by this network name. The tree name must be contained in double quotes.
	
	For example:
	
	`<action-triggers>`
	`<on-nds-login action-name="LoginInAction" tree="NCCD_TREE_1"/>`
	`</action-triggers>`

`on-ldap-login`	This element defines an action trigger that is activated when a user logs in to eDirectory through SecureLogin's LDAP client.
	
	Desktop Automation Services tests the primary connection to check whether the current server matches the server attribute specified in the configuration. If the current server matches the configuration, Desktop Automation Services executes the configured action identified by the action-name attribute value.
	
	`on-ldap-login` must be contained by an `action-triggers` parent element.
	
	`on-ldap-login` has two attributes:
	
	◆ **server:** The connections are tested periodically to test if the user is connected to a server matching this element name.
	◆ **action-name:** The name of an action defined in the input document that is executed when a user logs in to the tree named by the tree attribute.
	
	For Example:
	
	`<action-triggers>`
	`<on-ldap-login action-name="LoginInAction" server="192.168.1.255"/>`
	`</action-triggers>`
	
	NOTE: Ensure that the server address matches the LDAP Server address provided during installation.
	
	.

XML Tags	Description

`on-hot-key`	This element defines an action trigger to respond to the user typing the specified hot key sequence.This causes Desktop Automation Services to execute the matching action as defined in the input document. The `on-hot-key` elements must be contained within an `action-trigger` parent element.
	
	`on-hot-key` has three attributes:
	
	◆ **virtual-key:** The hex value of the key based on the virtual key map. This element specifies that it is the second component of the hot key sequence.
	◆ **modifiers:** The modifiers indicate the keys that are pressed in together with the virtual key to cause the hot-key event. The hex value might be a combination of one or more of the following, separated by a plus sign (+):
	◆ alt indicates the Alt key
	◆ ctrl indicates the Ctrl key
	◆ shift indicates the Shift key
	◆ win indicates the Windows key
	This element specifies that it is the first component of the hot key sequence.
	◆ **action-name:** The name of an action defined in the input document that is executed when the hot-key sequence is detected.
	
	For Example:
	
	`<action-triggers>`
	`<on-hot-key virtual-key="h" modifiers="ctrl+shift" action-name="HKeyAction"/>`
	`</action-triggers>`
	
	A virtual-key value of `'h'` and a modifiers value of ‘`ctrl+shift`’ produces a `Control-Shift-H` HotKey sequence.

XML Tags	Description

on-screen-saver	This element causes an action to be called when the workstation enters the screensaver mode. on-screen-saver elements must be contained by an action-trigger parent element.
	
	on-screen-saver has the following attributes:
	
	◆ **action-name:** The name of the action defined in the input document that is executed when the workstation has entered the screensaver mode and the specified interval has elapsed.
	◆ **interval:** The amount of time in milliseconds that the ARSControl waits before running the specified action after a screensaver event is triggered.
	
	NOTE: To activate this trigger, you must have a Windows system screen saver selected. Set the screen saver wait time to the desired time interval before the workstation activates the screen saver. If you are using DAS to activate the screen saver through the on-inactivity-timer action trigger, set the wait time to a longer timer interval than what you set for the on-inactivity-timer action trigger. For example, you can set the on-inactivity-timer interval to 60 minutes. The screen saver is triggered from DAS on the shared workstation.
	
	For Example:
	
	`<action-triggers>`
	`  <on-screen-saver action-name="logoff" interval="60000"/>`
	`</action-triggers>`
	
	This results in the logoff action being executed 60 seconds after the Windows screen saver is activated.

on-cardmon	The on-cardmon element specifies the action to be performed when a smart card is inserted, removed or when it is used for the login to the directory. If a user is logged in through a smart card and logs out because of a security reason, a specific action like a system lock must be performed to ensure that the workstation security is not at risk.
	
	The on-cardmon element must be contained within an action-trigger parent element.
	
	on-cardmon has the following attributes:
	
	◆ **action-name:** The name of the action defined in the input document that is executed when the card is removed from the reader.
	◆ **card-insert:** The name of the action defined in the input document that is executed when the card is inserted to the reader.
	◆ **LoginAction:** The name of the action defined in the input document that is executed when the user successfully logs into the Directory.
	
	If the login action fails card-insert action is repeated again.
	
	For example:
	
	`<action-triggers>`
	`  <on-cardmon action-name="Removal" card-insert="Insertion"`
	`LoginAction = "unHideMe" />`
	`</action-triggers>`

XML Tags	Description

on-Tap-cardmon	The on-Tap-cardmon element specifies the action to be performed when a contactless smart card is tapped on and tapped out, or when it is used for the login to the directory. If a user is logged in through a smart card and logs out because of a security reason, a specific action like a system lock must be performed to ensure that the workstation security is not at risk.
	
	The on-Tap-cardmon element must be contained within an action-trigger parent element.
	
	on-Tap-cardmon has the following attributes:
	
	◆ **action-name:** The name of the action defined in the input document that is executed when the card is tapped out from the reader.
	
	◆ **card-tapon:** The name of the action defined in the input document that is executed when the card is tapped on to the reader.
	
	◆ **LoginAction:** The name of the action defined in the input document that is executed when the user successfully logs into the Directory.
	
	If the login action fails card-tapon action is repeated again.
	
	◆ **TapCardSwitchUser:** This attribute is used to restrict the card tap to switch users in the kiosk mode. If this attribute value is set to true, then single card tap is required to switch the user in kiosk mode. If this attribute value is set to false, then double card tap is required to switch the user in kiosk mode.
	
	For example:
	
	`<action-triggers>`
	`  <on-Tap-cardmon action-name="Tappedout" card-tapon="Tappedon"`
	`LoginAction = "unHideMe" TapCardSwitchUser="true"/>`
	`</action-triggers>`

on-ad-login	This element defines an action trigger to poll for a user logging in to a workstation,in the Active Directory domain.
	
	This support is for SecureLogin installed in the Active Directory mode. If a user logs in to Active Directory, an action trigger invokes Desktop Automation Services which in turn executes the configured action identified by the action-name. on-ad-login element must be contained within the parent element, action-triggers.
	
	For example:
	
	`<action-triggers>`
	`  <on-ad-Login action-name="LogInAction"/>`
	
	`</action-triggers>`

XML Tags	Description

`action`	This is the parent element for all the commands that constitute an action.
	
	`action` has two attributes:
	
	name: The name can be any arbitrary string value. The character case in the name used by a caller to invoke an action must match the case used where the action is defined. The action-name must be contained in double quotes.
	
	multi-delay: This command element specifies the interval in executing the same action, twice.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <action name="ctrl+l" multi-delay="4000"> </action></application-runner-script>`
	
	`action` can contain any number of the following child elements:
	
	◆ Hide-Desktop and Unhide-Desktop
	◆ run-application
	◆ test-app-running
	◆ kill-app
	◆ kill-all-apps
	◆ map-drive
	◆ map-home-drive
	◆ map-location-drive
	◆ test-logged-in
	◆ test-ldap-logged-in
	◆ test-ad-logged-in
	◆ ad-logout
	◆ nds-logout
	◆ ldap-logout
	◆ screen-saver-on
	
	For example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <map-drive drive-letter="o:" remote-name="\\192.168.1.255\sys"/> </action></application-runner-script>`

XML Tags	Description
`Hide-Desktop` and `Unhide-Desktop`	The `Hide-Desktop` and `Unhide-Desktop` actions will hide or show the items on the user's desktop. These commands can be used with the on-login element to disable the users desktop prior to login and then to enable the desktop when the user login has been completed. **NOTE:** These actions are primarily for a kiosk approach without role-based access or for workstation policies managed through ZENworks® syntax. If you specify special characters in the `action.xml` file, it does not behave as expected.

`run-application`	This command element provides information that enables Desktop Automation Services to run an application and respond when the application is closed. There are four required attributes and one optional attribute.Following are the required attributes:
	
	◆ **application:** The name of the application to launch, such as `notepad.exe`. For applications that the operating system cannot find through the configured path environment variable, specify the complete application path and file extension.
	◆ **parameters:** Lists the required parameters to be passed to the application. This attribute should have text values enclosed within double quotes (“”).
	◆ **serial:** This attribute defines the application to run in synchronous or asynchronous mode. The attribute value can be either `true` or `false`. When this attribute is set to `true` (in synchronous mode), then the execution of the parent action does not continue until the application is closed or the interval timeout has expired.
	◆ **interval:** The timeout interval is used only used when the serial attribute is set to `true` (synchronous mode). If the application has not returned by the specified timeout, Desktop Automation Services stops waiting for a return and executes the next action.
	
	Following is an optional attribute:
	
	◆ **on-exit-action:** When the application started by this element is closed, the specified action is called.
	
	`run-application` cannot have any child elements.
	
	For example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<run-application application="C:\Program Files\Mozilla Firefox\firefox.exe" parameters="" on-exit-action="launchSomethingElseAction"serial="true" interval="500"/>`
	`</action>`
	`</application-runner-script>`

XML Tags	Description
`test-app-running`	The `test-app-running` command element provides information that enables Desktop Automation Services to test whether an application is running or not. . `test-app-running` can have only one attribute: ◆ **application:** The name of the application as it is found in the process list. Because `test-app-running` is a test command, it can contain either one or both of the following child elements: ◆ **if-true:** An element containing the command operations to perform if the test returns a true value. ◆ **if-false:** An element containing the command operations to perform if the test returns a false value. For example: `<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <test-app-running application="notepad.exe"> <if-true> <kill-app application="xmlspy.exe"/> <kill-all-apps exclude-apps="notepad.exe:xmlspy.exe"/> <map-drive drive-letter="F:" remote-name="\\172.16.5.250\sys"/> </if-true> <if-false> <map-drive drive-letter="G:" remote-name="\\192.168.1.255\sys"/> </if-false> </test-app-running> </action></application-runner-script>`
`kill-app`	The `kill-app` command element provides information that enables Desktop Automation Services to close an application. `kill-app` has one required attribute and one optional attribute. Following is the required attribute: ◆ **application:** The name of the application to close, as found in the process list. Following is the optional attribute: ◆ **interval:** The amount of time in milliseconds that Desktop Automation Services waits after sending a close command to the application before killing the process. The default interval value is 1000. `kill-app` cannot contain any child element. For Example: `<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <kill-app application="xmlspy.exe"/> </action></application-runner-script>`

XML Tags	Description
`kill-all-apps`	This command element provides information that enables Desktop Automation Services to kill all the running applications except those specified in exclude-apps. `kill-all-apps` has one required attribute and one optional attribute. Following is the required attribute: ◆ **exclude-apps:** The names of the applications that must not be killed. The application names are separated by a colon (:) character. The name of an application listed in this attribute must match the name of the application listed in the **Processes** tab of the Task Manager. Following is the optional attribute: ◆ **interval:** The amount of time in milliseconds that Desktop Automation Services waits after sending a close command to an application before killing the process. Because each process is closed in a sequential order, a large interval significantly increases the amount of time the command takes to execute. The default value is 0. `kill-all-apps` cannot have any child elements. For Example: `<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <kill-all-apps exclude-apps="notepad.exe:xmlspy.exe"/> </action></application-runner-script>`
`map-drive`	This command element enables Desktop Automation Services to perform a drive mapping. `map-drive` has two required attributes: ◆ **drive-letter:** Specifies the drive letter to assign to the new mapped drive. ◆ **remote-name:** Specifies the path in UNC format for a remote volume to be mapped. `map-drive` cannot contain child elements. For example: `<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <map-drive drive-letter="G:" remote-name="\\192.168.1.255\sys"/> </action></application-runner-script>`

XML Tags	Description
`map-home-drive`	This command element enables Desktop Automation Services to map a drive to a home directory as defined by the homedrive attribute in the user's directory object. `map-home-drive` has two required attributes: ◆ **drive-letter:** Specifies the drive letter to assign to the new mapped drive. This value should be a letter representing the drive pointer, followed by a colon (:) character enclosed within double quotes. ◆ **tree:** Specifies the tree containing the object with the home directory information. `map-home-drive` cannot contain any child elements. For Example: `<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <map-home-drive drive-letter="I:" tree="TestTree"/> </action></application-runner-script>`
`map-location-drive`	This command element enables Desktop Automation Services to map a drive based on a properties file. The `properties` file is an acscii based text file that contains the necessary mapping information for one or more drives. `map-location-drive` has four required attributes: ◆ **drive-letter:** Specifies the drive letter to assign to the new mapped drive. This value should be a letter representing the drive pointer followed by a colon (:) character enclosed within double quotes. ◆ **tree:** Specifies the tree containing the object with the location information. ◆ **attribute:** Specifies the key to be used to obtain a value from the properties file. ◆ **file-name:** Specifies the file system path to a properties file containing information for the `map-location-drive` operation. This file contains property information in the form of key or value pairs. The property key is located on the left of the equals symbol (=) and the value is on the right side. For example: here=\\137.65.60.39\Share2 there=\\137.65.60.39\Share3 `map-location-drive` cannot contain any child elements. For Example: `<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script> <action name="sample-action"> <map-location-drive drive-letter="T:" tree="TestTree2" file-name="c:\yourFile.c" attribute="yourAttribute"/> </action></application-runner-script>`

XML Tags	Description

`test-logged-in`	This command element provides information that enables Desktop Automation Services to test whether the user is logged in to a particular eDirectory server or not.
	
	`test-logged-in` has one required attribute:
	
	◆ **tree:** The name of the tree for which the logged in state has to be tested.
	
	Because the `test-logged-in` is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<test-logged-in tree="TestTree">`
	`<if-true>`
	`<run-application application="explorer.exe" parameters="" serial="false" interval="1000"/>`
	`<map-home-drive drive-letter="I:" tree="TestTree"/>`
	`</if-true>`
	`<if-false>`
	`<map-location-drive drive-letter="J:" tree="TestTree" file-name="c:\myFile.c" attribute="myAttribute"/>`
	`</if-false>`
	`</test-logged-in>`
	`</action>`
	`</application-runner-script>`

XML Tags	Description

`test-ldap-logged-in`	This command element provides information that enables Desktop Automation Services to test whether the user is logged in to a particular LDAP server or not. This command must only be used when using the LDAP GINA/Credential Provider for authentication.
	
	`test-ldap-logged-in` has one required attribute:
	
	◆ **server:** The name of the server for which the logged-in state must be tested.
	
	Because `test-ldap-logged-in` is a test command, it can contain either or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	For example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<test-ldap-logged-in server="192.168.1.255">`
	`<if-true>`
	`<run-application application="explorer.exe" parameters="" serial="false" interval="1000"/>`
	`</if-true>`
	`<if-false>`
	`<run-application application="iexplore.exe" parameters="" serial="false" interval="1000"/>`
	`</if-false>`
	`</test-logged-in>`
	`</action>`
	`</application-runner-script>`

`test-ad-logged-in`	The `test-ad-logged-in` command element provides information that enables Desktop Automation Services to test whether a user is logged into the Active Directory
	
	`test-ad-logged-in` contains no attributes. It contains either one or both of the following child elements:
	
	◆ **if-true:** This element contains the command operations to perform if the test returns a true value.
	
	◆ **if-false:** This element contains the command operations to perform if the test returns a false value
	
	For Example:
	
	`<action name="test">`
	`<test-AD-logged-in>`
	`<if-true>`
	`<message-box caption="User are logged in" />`
	`</if-true>`
	`<if-false>`
	`<message-box caption="User are not logged in" />`
	`</if-false>`
	`</test-AD-logged-in>`
	`</action>`

XML Tags	Description

ad-logout	This test command element provides information that enables Desktop Automation Services to log out of the SecureLogin in the Active Directory mode.
	
	ad-logout does not have any attributes or child attributes.
	
	For Example:
	
	`<application-runner-script>`
	`  <action name="sample-action">`
	`  <AD-logout/>`
	`  </action>`
	`</application-runner-script>`

nds-logout	This test command element provides information that enables Desktop Automation Services to log out of the primary NDS® connection.
	
	nds-logout has no attributes.
	
	nds-logout has no child attributes.
	
	For example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`  <action name="sample-action">`
	`  <nds-logout/>`
	`  </action>`
	`</application-runner-script>`

ldap-logout	This test command element provides information that enables Desktop Automation Services to log out of SecureLogin.
	
	ldap-logout can have one optional attribute:
	
	◆ **gina:** Can have either true or false values. If the value is true, the login dialog box for SecureLogin is displayed after logging out of SecureLogin. If the value is false, no action is taken.
	The default value is true.
	
	ldap-logout cannot have any child elements.
	
	For example:
	
	`<action name="logoff">`
	`  <pause interval="100"/>`
	`  <kill-all-apps exclude-apps="slbroker.exe:slwinsso.exe:slproto.exe:explorer.exe:"/>`
	`  <ldap-logout gina="true"/>`
	`</action>`

XML Tags	Description

`screen-saver-on`	This action tag invokes the Windows screen saver, which triggers the on-screen-saver action. When this action is triggered, the Windows screen saver is started and the DAS `on-screen-saver` is invoked with timer.
	
	This action locks the workstation and triggers the screen saver, which covers up any icons and browsers. `screen-saver-on` elements must be contained by an `action-triggers` parent element.
	
	Use Case: A user is away from the workstation. A smartcard device triggers an event to start the Windows screen saver program. After the defined time interval of inactivity, the user is logged out. If an activity occurs, the screen saver closes; the user is not logged out. The user returns to the workstation, which is in an undisturbed state. The screen-saver-on action ensures that the icons and browsers are covered.
	
	`screen-saver-on` has one optional attribute:
	
	◆ **lock:** If lock is set to true, the workstation is locked after the screen saver is activated. The user must enter the password to unlock the workstation and the screen saver.
	
	If lock is set to false, the workstation lock is not activated. Any mouse movement or keystroke deactivates the screen saver.
	
	For Example:
	
	`<action name="Act1">`
	`<screen-saver-on/>`
	`</action>`
	`<action name="Act2">`
	`<screen-saver-on lock="true"/>`
	`</action>`
	`<action name="Act3">`
	`<screen-saver-on lock="false"/>`
	`</action>`
	`<action-triggers>`
	`<on-hot-key virtual-key="l" modifiers="ctrl" action-name="Act1"/>`
	`<on-hot-key virtual-key="m" modifiers="ctrl" action-name="Act2"/>`
	`<on-hot-key virtual-key="n" modifiers="ctrl" action-name="Act3"/>`
	`</action-triggers>`

XML Tags	Description

`test-nds-attr-val`	This test command element provides information that enables Desktop Automation Services to test whether or not the currently logged in users NDS account contains a particular directory attribute with a particular value.
	
	`test-nds-attr-val` has four required attributes:
	
	◆ **tree:** The name or IP address of the tree containing the user account to be searched for the attribute value.
	
	◆ **attr-name:** The name of the attribute to be tested in the NDS account.
	
	◆ **attr-syntax:** The syntax of the attribute to be tested in the NDS account.
	
	The acceptable `attr-syntaxes` are:
	
	◆ string
	
	◆ integer
	
	◆ boolean
	
	◆ **attr-val:** The value to be searched in the target attribute in the NDS account. The values for the Boolean syntax attribute must be either true or false.
	
	NOTE: If the attribute syntax is string, then the comparison between the value retrieved from the eDirectory and the value of the attr-val is case sensitive.
	
	Because the `test-nds-attr-val` is a test command, it can contain either or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action1">`
	`<test-nds-attr-val tree="TestTree" attr-name="cn" attr-syntax="string" attr-val="larry">`
	`<if-true>`
	`<kill-app application="george.exe"/>`
	`<run-application application="fred.exe" parameters="" serial="true" interval="250"/>`
	`</if-true>`
	`<if-false>`
	`<map-drive drive-letter="S:" remote-name="\\172.16.5.253\sys"/>`
	`</if-false>`
	`</test-nds-attr-val`
	`</action>`
	`<action name="sample-action2">`
	`<test-nds-attr-val tree="TestTree" attr-name="Password Minimum Length" attr-syntax="integer" attr-val="5">`
	`<if-true>`
	`<!â€”any commands may be inserted here-->`
	`</if-true>`
	`<if-false>`
	`<!â€"any commands may be inserted here-->`
	`</if-false>`
	`</test-nds-attr-val>`
	`</action>`
	`<action name="sample-action3">`
	`<test-nds-attr-val tree="TestTree" attr-name="Password Required" attr-syntax="boolean" attr-val="true">`
	`<if-true>`
	`<!â€"any commands may be inserted here-->`
	`</if-true>`
	`<if-false>`

XML Tags	Description

`test-ip-subnet`	This test command is useful for enabling an action to determine if the workstation resides on a particular network or not. This can be critical if the action is deciding whether to launch a particular application that is available or effective in a given network.
	
	When invoked, the `test-ip-subnet` command executes the child commands if the current subnet of the workstation and the command's `addr` attribute value are the same.
	
	`test-ip-subnet` has two required attributes:
	
	◆ **addr:** An IP subnet to compare with the local IP addresses of the machine.
	◆ **subnet:** The subnet mask (in the form of 255.255.255.0) is applied to the addr attribute and the local IP addresses, which are then compared. If the network portion matches, the test returns a true value.
	
	Because the `test-ip-subnet` is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<test-ip-subnet addr="192.168.1.0" subnet="255.255.255.0">`
	`<if-true>`
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	`</if-true>`
	`<if-false>`
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	`</if-false>`
	`</test-ip-subnet>`
	`</action>`
	`</application-runner-script>`

XML Tags	Description

`test-env-variable`	This test command element enables Desktop Automation Services to test whether an environment variable matches a specific value or not.
	
	`test-env-variable` has two required attributes:
	
	◆ **var-name:** The case-sensitive environment variable name. If the variable does not exist, the test returns a false value.
	◆ **var-value:** The value used for case-insensitive comparison with the actual variable value.
	
	Because `test-env-variable` is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<test-env-variable var-name="Testvar" var-value="testvalue">`
	`<if-true>`
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	`</if-true>`
	`<if-false>`
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	`</if-false>`
	`</test-env-variable>`
	`</action>`
	`</application-runner-script>`

`message-box`	This command element provides information that enables Desktop Automation Services to display a message box.
	
	`message-box` has two required attributes:
	
	◆ **caption:** The text to be displayed in the dialog box.
	◆ **window-name:** The title for the dialog box window.
	
	`message-box` does not have any child elements.
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<message-box caption="HotKey Control+H was pressed." window-name="HotKey Event"/>`
	`</action>`
	`</application-runner-script>`

XML Tags	Description

execute-user-action	The value of this attribute can be set using iManager or other equivalent eDirectory management tool. To edit the value of this attribute with iManager, open the properties of the user object and then navigate to the other tab. Find the attribute ARSUserConfiguration and edit the value. The value must be formatted in XML syntax as used by the Desktop Automation Services.
	
	NOTE: The XML information stored in the user object can contain only actions. Triggers are not supported.
	
	execute-user-action has one required attribute:
	
	◆ **action-name:** The name of the configured action read from the user object.
	
	Example value for the ARSUserConfiguration attribute:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script><action name="userAction"><!--.  Any actions may be inserted here. . --></action></application-runner-script>`
	
	execute-user-action Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"><application-runner-script><action name="sample-action"><execute-user-action action-name="userAction"/></action></application-runner-script>`

XML Tags	Description

`if-true`	This is one of the two allowed types of child elements for a test type of command. The other element is "if-false" on page 156.
	
	`if-true` contains the result of all the test commands that return a true value. So, `if-true` can be a parent element for all the commands that constitute an action.
	
	`if-true` does not have any attribute values.
	
	`if-true` can contain any number of the following child elements:
	
	◆ `run-application`
	◆ `test-app-running`
	◆ `kill-app`
	◆ `kill-all-apps`
	◆ `map-drive`
	◆ `map-home-drive`
	◆ `map-location-drive`
	◆ `test-logged-in`
	◆ `test-ldap-logged-in`
	◆ `test-nds-attr-val`
	◆ `test-ip-subnet`
	◆ `test-env-variable`
	◆ `message-box`
	◆ `nds-logout`
	◆ `ldap-logout`
	◆ `execute-user-action`
	
	For Example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<test-env-variable var-name="Testvar" var-value="testvalue">`
	`<if-true>`
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	`</if-true>`
	`<if-false>`
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	`</if-false>`
	`</test-env-variable>`
	`</action>`
	`</application-runner-script>`

XML Tags	Description

`if-false`	This is one of the two allowed types of child elements for a test type of command. The other element is “if-true” on page 155.
	
	`if-false` contains the result of all the test commands that return a false value. So, `if-false` can be a parent element for all the commands that constitute an action.
	
	`if-false` does not have attribute value.
	
	`if-false` can contain any number of the following child elements:
	
	◆ `run-application`
	◆ `test-app-running`
	◆ `kill-app`
	◆ `kill-all-apps`
	◆ `map-drive`
	◆ `map-home-drive`
	◆ `map-location-drive`
	◆ `test-logged-in`
	◆ `test-ldap-logged-in`
	◆ `test-nds-attr-val`
	◆ `test-ip-subnet`
	◆ `test-env-variable`
	◆ `message-box`
	◆ `nds-logout`
	◆ `ldap-logout`
	◆ `execute-user-action`
	
	For example:
	
	`<?xml version="1.0"?><!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	`<action name="sample-action">`
	`<test-env-variable var-name="Testvar" var-value="testvalue">`
	`<if-true>`
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	`</if-true>`
	`<if-false>`
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	`</if-false>`
	`</test-env-variable>`
	`</action>`
	`</application-runner-script>`

Using the DAS Editor to Configure Actions

Each instance of DAS is driven by an XML document describing the actions that are available.

This release of SecureLogin introduces a DAS Editor that helps you in composing an action. The wizard tool is available in \SecureLogin\Tools\ DAS Editor folder on the SecureLogin CD image.

The DAS Wizard provides an ability to create and modify a configuration file. The wizard helps you create the configuration with the correct XML format.

- If you are using the DAS Wizard to create configuration file, select the actions and triggers applicable for DAS.
- The DTD file must be stored in the same location as the XML file. The DTD file validates the XML file.
- “Creating a New Configuration File” on page 157

Creating a New Configuration File

1. Run the `DASWizard.exe` available in the `\SecureLogin\Tools\DAS Editor` folder of the SecureLogin executables. The wizard is launched.
2. Right-click on the action header, and select **Insert Action**.
3. Specify a name for the action and press Enter. For example, WSAction.
4. Define a child element for the action that was created.

   Right-click the action you added, click **Insert Element**, and select the child element. For example, map-drive.

   Note that the Editor will only list the DAS actions that are valid for child elements.

   NOTE: The Editor lists only the DAS actions that are valid for child elements.

   ---

5. Specify the attributes for the child element you defined in Step 4. Note that not all of the child elements have attributes. For example, `nds-logout` or `show-desktop` do not have any required attributes.

   For example, if you selected map-drive in Step 4, then specify the drive letter and the UNC path to be mapped.
6. To add a trigger to the action you have created, right-click on the header trigger, followed by selecting Insert trigger. The Editor will display a list of valid triggers that can be selected. Select the desired trigger.
7. Specify the attributes for the trigger you selected.
8. The configuration specified is saved to a new text file in the required XML format.

   The DTD file must be copied to the same location as the xml file that you want to edit. If you create a new configuration file by following the previous steps and require to open the file for further editing, then you need to copy the DTD file to the same location. Creating a new configuration file uses the DTD file from the DAS Editor folder for validation of the XML syntax. If the DTD file is not located the following error message is displayed:

   ```
   Error Loading XML
   Error 0x800c0006 on line x, position x
   Reason: The system cannot locate the object specified.
   Error processing resource 'ARS_1.0.dtd'.
   ```

Example of an Action File

This example XML file contains examples of most of the XML elements that can be used to compose action sequences in Desktop Automation Services.

```
<?xml version="1.0"?>
<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">
<application-runner-script>
  <action name="worksuite">

    <!-- KILL THE GAMES -->
    <kill-app application="freecell.exe"/>
    <kill-app application="winmine.exe"/>
    <kill-app application="sol.exe"/>

    <!-- LOAD THE WORK APPS -->
    <test-app-running application="notepad.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="notepad.exe" on-exit-action="gamesuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="calc.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="calc.exe" on-exit-action="gamesuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="mspaint.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="mspaint.exe" on-exit-action="gamesuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
  </action>
  <action name="gamesuite">

    <!-- KILL THE WORK APPS -->
    <kill-app application="notepad.exe"/>
    <kill-app application="calc.exe"/>
    <kill-app application="mspaint.exe"/>

    <!-- LOAD THE GAMES -->
    <test-app-running application="freecell.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="freecell.exe" on-exit-action="worksuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="winmine.exe">
      <if-true>
```

```
      </if-true>
      <if-false>
        <run-application application="winmine.exe" on-exit-action="worksuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="sol.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="sol.exe" on-exit-action="worksuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
  </action>
</application-runner-script>
```

NOTE: Additional action file examples can be found at `SecureLogin\Tools\DAS Editor\Script Samples\XML` on the SecureLogin CD image.

Usage Scenario

This section explains the SecureLogin configuration that is required to switch user using Desktop Automation service and Advanced Authentication.

To tap smart card to switch user, perform the following:

1. Install SecureLogin with Advanced Authentication and Desktop Automation Service (DAS)
2. Perform the following to configure Kiosk mode in SecureLogin.
   1. On the Windows desktop, click **start > Run** to display the Run dialog box.
   2. Enter `regedit,` then click **OK** to open the Registry Editor.
   3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\NSLADAuth`.
   4. Create `DWORD NSLADAuth` and set the value of `NSLADAuth` to 1.
3. Edit the action.xml file and set the value of the attribute `TapCardSwitchUser` to `true` to switch the user in single card tap.

   The DAS action.xml file is located at `C:\Program Files\NetIQ\SecureLogin\Desktop Automation Services\actions.xml`.

   A sample action.xml file with `TapCardSwitchUser` attribute is as follows:

```
<?xml version="1.0"?>

<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">

<!-- KP Base Windows Action for Active Directory Mode Version: 1.02 -->

<!-- Inactivity Counter is supposed to be working -->

<application-runner-script>

<action name="startup">

<test-app-running application="sltray.exe">
```

```
<if-true>

<AD-logout gina="false" />

<!-- delay for NSL to successfully shutdown -->

<pause interval="750" />

<hide-desktop/>

<pause interval="750" />

<!-- <kill-all-apps exclude-
apps="slproto.exe:slwinsso.exe:slbroker.exe:explorer.exe:notepad.exe" /> -->

<pause interval="750" />

<run-application application="sltray.exe" parameters="" on-exit-action=""
serial="true" interval="500"/>

</if-true>

<if-false>

<hide-desktop />

<pause interval="750" />

<run-application application="sltray.exe" parameters="" on-exit-action=""
serial="true" interval="500"/>

</if-false>

</test-app-running>

</action>

<action name="showdesktop">

<unhide-desktop/>

</action>

<action name="SCLogoff">

<AD-logout gina="false" />

<!-- delay for NSL to successfully shutdown -->

<pause interval="750" />

<hide-desktop/>

<pause interval="750" />

<!-- <kill-all-apps exclude-
apps="slproto.exe:slwinsso.exe:slbroker.exe:explorer.exe:notepad.exe" /> -->

<pause interval="750" />
```

```
<run-application application="sltray.exe" parameters="" on-exit-action=""
serial="true" interval="500"/>

</action>

<action name="insert">

<test-app-running application="sltray.exe">

<if-true></if-true>

<if-false>

<run-application application="sltray.exe" parameters="" on-exit-action=""
serial="true" interval="500"/>

</if-false>

</test-app-running>

</action>

<action-triggers>

<on-Tap-cardmon action-name="SCLogoff" card-tapon="insert" LoginAction=
"showdesktop" TapCardSwitchUser="true"/>

</action-triggers>

</application-runner-script>
```

4 Perform the following to configure DAS to load on startup:

1. On the Windows desktop, click **start > Run** to display the Run dialog box.
2. Enter `regedit,` then click **OK** to open the Registry Editor.
3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`.
4. Create a String with any name and set the path to DAS executable as value. For example: `DAS : C:\Program Files\NetIQ\SecureLogin\Desktop Automation Services\ARS.exe startup`

   NOTE: Startup is the additional parameter used in DAS to invokes a default action defined in the actions.xml during Windows startup.

5 Reboot the system and perform single tap to switch user.

21 LDAP SSL Server Certificate Verification

This section contains the following information:

About LDAP SSL Server Certificate Verification

The LDAP SSL server certificate verification is a security feature that was introduced in the SecureLogin 6.0 SP1 release. This feature allows the client to verify the trustworthiness of the server, using a process similar to the certificate verification process carried out by browsers like Microsoft Internet Explorer and Mozilla Firefox.

Certificate verification of the server is important to prevent potential security risks. It is essential that the client verify the server certificate during the LDAP SSL connection to the server. If the client cannot verify the server certificate, it is possible that an intruder on the same subnet can decrypt the communication between the client and access user credentials.

By default, eDirectory is configured for self-signed certificates. Although self-signed certificate works, it does not pass all the validation checks carried out during the verification process. So, users are prompted to validate the certificate at the first time they attempt to access the server. To prevent this, you can obtain a signed certificate from a known certificate authority such as VeriSign and replace the existing certificate.

Validating an LDAP SSL Server Certificate

During the establishment of an LDAP SSL connection, client receives the root certificate from the server so that the client can verify the trustworthiness of the server. The client uses the following process to validate the certificate:

- It compares the current certificate with any of the previously stored certificate. If the certificates match, the client does not perform further checks, and adds the certificate to the local store. If the certificates do not match, the client continues the validation process.
- It checks whether the certificate is trusted. This ensures that a known authority is issuing the certificate.
- It checks whether the date on the certificate is valid with reference to the current date.
- It checks whether the host name on the certificate matches the date on the server.

If the certificate passes these preceding tests, the client adds the certificate to local store so it can be used for future verification.

If the certificate does not pass the verification process, the application prompts you to either continue the connection or terminate the connection.

- To continue the connection, click **Yes**. The certificate is added to the local store so it can be used for future verification, and the authentication process continues.

- To terminate the connection, click **No**.
- To get details about the certificate, click **View Certificate** to display the Certificate Information dialog box shown in the proceeding figure. If you decide that the certificate is valid, you can click **Install Certificate** to permanently install the certificate.

NOTE: The Windows workstation local store is different from the SecureLogin LDAPAuth clients certificate store.

Enabling LDAP SSL Certificate Verification

By default, the certificate verification feature is disabled. You can enable this feature by adding the following registry value:

1. On a Windows workstation, click **Start > Run** to display the Run dialog box.
2. Type `regedit` then click **OK** to open the Registry Editor.
3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Login\LDAP` registry hive.
4. Create a new DWORD Value named `VerifySSLCert` and set it to 1.
5. Exit the Registry Editor.

22 Security Considerations

Consider the following to help ensure security for SecureLogin:

- Use the AES (Advanced encryption standard) or Triple DES (Data Encryption Standard) for the encryption of SecureLogin data.
- Back up SecureLogin data and directory data by using encryption and password protection.
- Use `AAVerify` to provide additional advanced authentication to single sign-on applications with NMAS methods or other AA methods such as NetIQ Advanced Authentication Framework.
- Implement smart cards, storing application credentials on cards and the encryption of the data store using PKI credentials.
- Protect the SecureLogin desktop shortcut with a password so that others cannot view SecureLogin data.
- Prevent certain SecureLogin settings and options from being visible or modifiable by others.
- Use a universal password for increased security by providing additional layers of policies.
- Require SecureLDAP when using LDAP to authenticate to SecureLogin.
- Use Novell SecretStore to provide additional security to SecureLogin data stored on eDirectory.
- Use AA methods such as OTP and NMAS to provide advanced authentication, such as fingerprint, and token-based authentication.
- Store SecureLogin credentials in a PIN-protected smart card, which provides a secure, portable, and efficient single sign-on solution.
- Keep the local cache files in a user profile directory so that only the corresponding Windows user can access them.
- Enable a passphrase to provide additional security to SecureLogin user data.
- Ensure strict password policies for SecureLogin users and for all single sign-on logins. Randomization of passwords and hiding them from end users is also essential.
- Use auditing features such as NetIQ Sentinel, SNMP alerts and Windows event logs to capture SecureLogin activity wherever applicable.
- When you are using LDAP with NMAS, the SecureLogin universal password must be enabled.

23 SecureLogin Security Role Configuration for Active Directory

For a user to administer SecureLogin in an Active Directory environment, the user must have both sufficient permissions to the SecureLogin attributes and the SecureLogin settings to allow the users proper access.

Directory Attributes

The protocom attributes hold user or container data that is used by SecureLogin to provide Single Sign-On functionality. These attributes are named as follows:

protocom-SSO-Auth-Data
protocom-SSO-Entries
protocom-SSO-Entries-Checksum
protocom-SSO-Profile
protocom-SSO-Security-Prefs
protocom-SSO-Security-Prefs-Checksum

The function for each of these attributes is as follows:

protocom-SSO-Auth-Data:

- This attribute is only for a User object. It is an octet-string type.
- It contains all user-specific authentication data, such as the passphrase.

protocom-SSO-Entries:

- This attribute is for User, Container, and Organizational Unit objects. It is an octet-string type. This attribute contains the following:
- All the user's login user IDs and passwords
- Specific preferences and application definitions at the User object
- Corporate application definitions and preferences at the Container and Organizational Unit objects

protocom-SSO-Entries-Checksum:

- This attribute optimizes the loading of data from the Directory. Whenever data changes in the protocom-SSO-Entries attributes, the Checksum attribute is updated. When SecureLogin loads, it reads the checksum and compares it to the checksum in memory. If the checksums are different, SecureLogin reloads the Entries attribute from the directory.

NOTE: This attribute is used only when the database mode is set to 6.0 or higher.

protocom-SSO-Profile:

- This attribute is used to instruct SecureLogin to read the settings and preferences from another container.

protocom-SSO-Security-Prefs:

- This attribute stores data required for SecureLogin to operate before loading the users datastore. This data can include Administrator-set Passphrase questions, Passphrase help information, settings, and similar things.

protocom-SSO-Security-Prefs-Checksum:

- This attribute functions with the protocom-SSO-Security-Prefs attribute much like the protocom-SSO-Entries-Checksum functions with the protocom-SSO-Entries attribute.

NOTE: This attribute is used only when the database mode is set to 6.0 or higher.

Directory Permissions Assignment

Based upon the above attribute descriptions and functions, specific roles might be granted the following permissions:

- Complete SecureLogin Management:
  - protocom-SSO-Auth-Data = Read and Write
  - protocom-SSO-Entries = Read and Write
  - protocom-SSO-Entries-Checksum = Read and Write
  - protocom-SSO-Security-Prefs = Read and Write
  - protocom-SSO-Security-Prefs-Checksum = Read and Write
- Script, Credentials, and Clear Object Data administration:
  - protocom-SSO-Auth-Data = Read and Write
  - protocom-SSO-Entries = Read and Write
  - protocom-SSO-Entries-Checksum = Read and Write

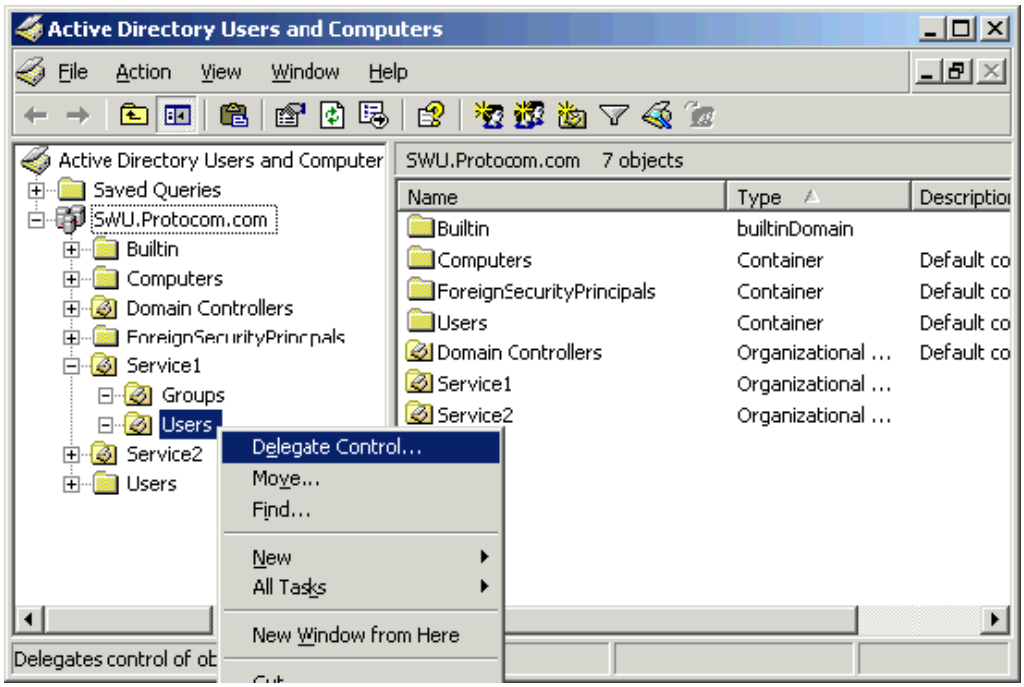
Depending on the needs of your organization, these permissions can be assigned to specific users or groups at an organizational unit level. The following discussion demonstrates the creation of a SecureLogin Administration group and the delegation of permissions to an organizational unit that is one level below the top level organizational units in the Directory hierarchy.

Assigning Permissions for SecureLogin Administrators

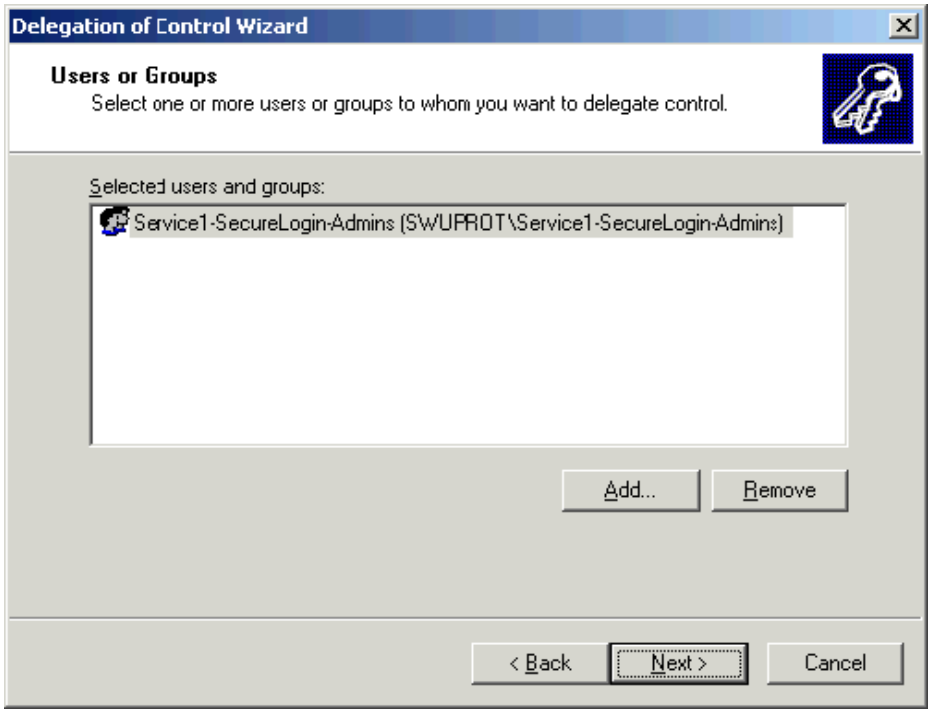
1. Login to the Active Directory domain as an administrative level user.
2. On a workstation or server, open **Active Directory User and Computers** (dsa.msc), and browse to the OU where you would like to create the group that will manage SecureLogin for the selected container and its children.
3. Click the create group button.
4. Give the group a descriptive name, such as Service1-SecureLogin-Admins.

5 Add the appropriate users to the group.

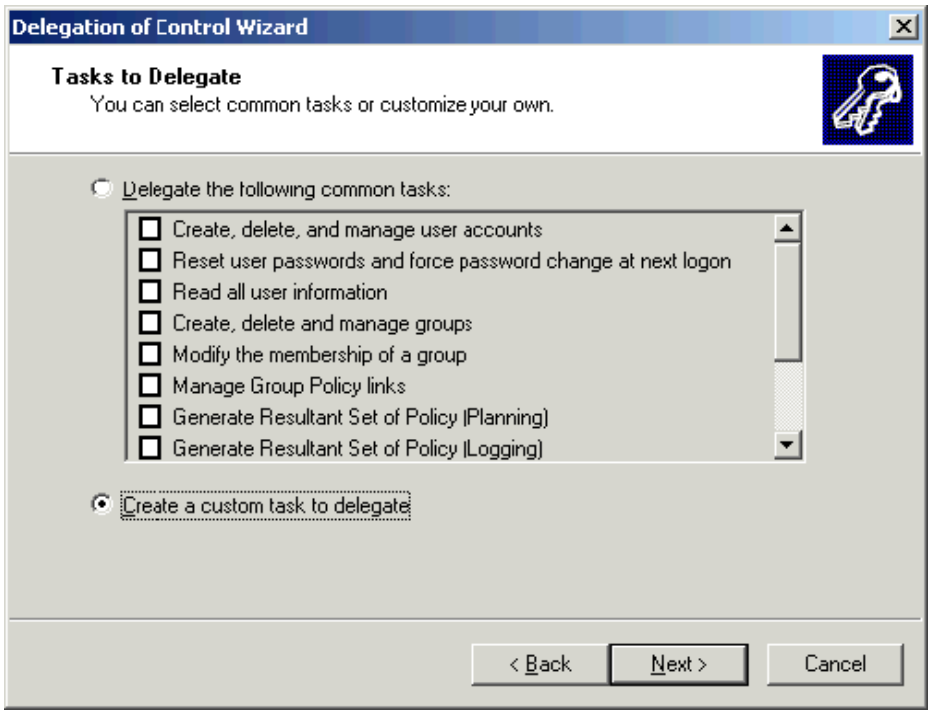
6 Delegate the permissions to the SecureLogin attributes at the container where the users are.



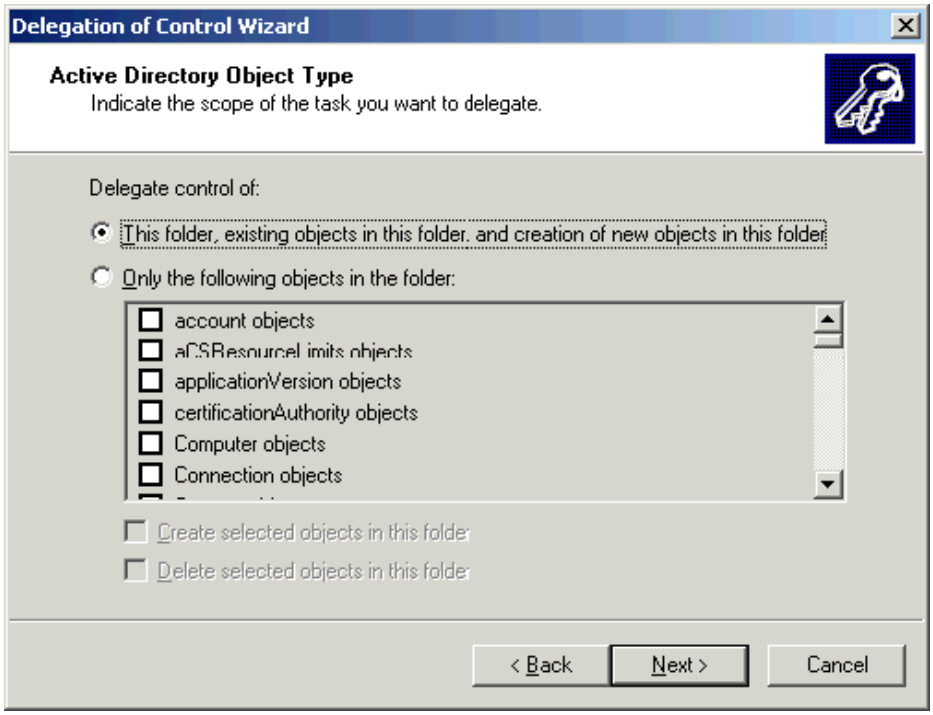
7 Add the group you want to delegate control, then click **Next**.



8 Select create a custom task to delegate, then click **Next**.



9 Select **This folder, existing objects in this folder, and creation of new objects in this folder,** then click **Next**.



10 Since these are administrator level users, they will be granted permissions to manage all aspects of the container and its subordinate objects. Select the **General**, **Property-specific** check boxes. Select the **Read**, **Write**, **Read All Properties**, and **Write All Properties** permissions.

Verify that you have all Protocom permissions with **Read** and **Write**. Click **Next** to continue.

11 You are now finished with the delegate control wizard for the Service1-SecureLogin-Admins group. Click **Finish**.

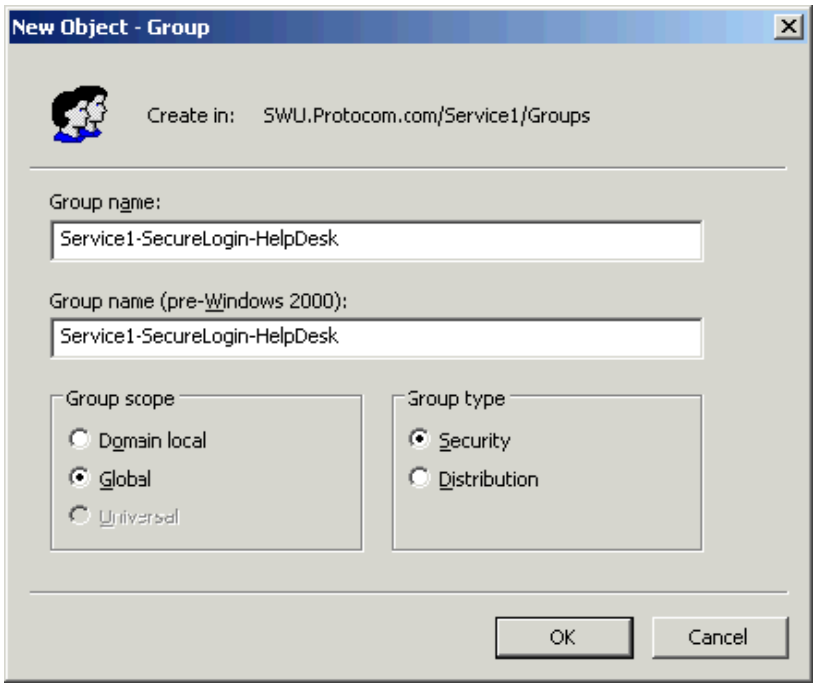
Assigning Permissions for SecureLogin Help Desk

1 Login to the Active Directory domain as a administrative level user.

2 On a workstation or server open Active Directory User and Computers, and browse to the OU where you would like to create the group that will hold the Help Desk users who will work with SecureLogin for the selected container and its children.

3 Click the create group button 

4 Give the group a descriptive name, such as Service1-SecureLogin-Help Desk.



5 Add the appropriate users to the group.

6 Delegate the permissions to the SecureLogin attributes at the container where the users are.

7 Add the group you want to delegate control to, then click **Next**.

8 Select create a custom task to delegate, then click **Next**.

9 Select the **Only the following objects in the folder**, then scroll down to user objects and select it. Click **Next**.

10 Since these are SecureLogin Help desk level users they will only be granted permissions to manage the SecureLogin attributes. Select the General and Property-Specific checkboxes. Then scroll down and select both the read and write permissions for all protocom- attributes.

11 You are now finished with the delegate control wizard for the Service1-SecureLogin-Admins group. Click **Finish**.

Assigning SecureLogin Client Settings for Administrators and Help Desk Groups

Now that you have assigned the correct Directory permissions to allow members of the administrators and help desk groups to read and write the protocom attributes, you need to assign the SecureLogin client settings (SecureLogin preferences) to allow them to see what they have permissions to access. This is required to override the more restrictive settings the user will inherit from their parent container.

To accomplish this, you can either directly modify the users individual settings. A viable approach if you have a few users who will be granted the elevated permissions. This said, many customers still choose the direct assignment approach, as it can reduce the steps when troubleshooting where someone is getting a specific client setting from. Alternatively, you might utilize SecureLogin's support for group policies. In either case, please see step 8 in this section of the document for the recommended settings.

For the sake of this document, it will be assumed you know how to assign individual user's settings, and thus this document will focus on the use of group policies (assuming the feature was enabled during the product installation). As stated previously, both methods have their merits and should be evaluated before deciding on an approach.

Creating the Group Policy

1. Login to the Active Directory domain as a administrative level user.
2. On a workstation or server open Active Directory User and Computers, and browse to the OU that contains the groups that you created earlier. Right click it, select **Properties**.
3. In the properties dialog that opens up, select the **Group Policy Tab**.

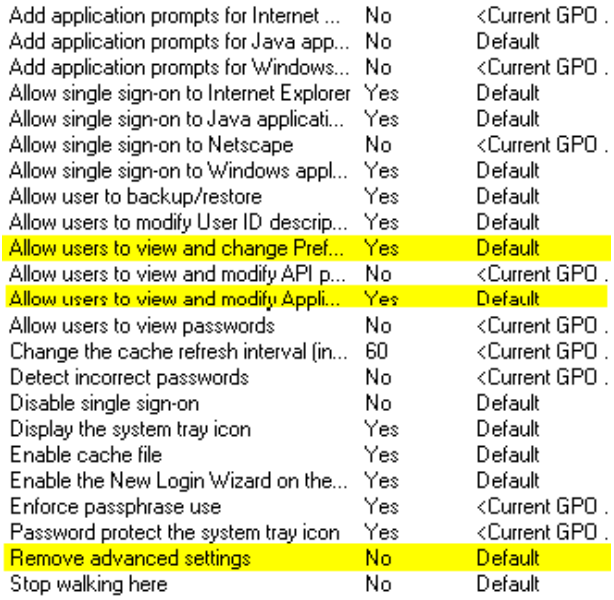
   NOTE: In this example the Group Policy Management snap-in has been installed. It can be downloaded from Microsoft (http://www.microsoft.com/downloads/details.aspx?FamilyID=0a6d4c24-8cbd-4b35-9272-dd3cbfc81887&displaylang=en)

   ---

4. Click the **Open** button, the Group Policy Management (GPM) interface will open. Select the **Group Policy Objects** container and right click it. Select **New**.
5. Enter a name for the GPO.
6. Right click the new GPO and select **Edit**.
7. Browse to the **User Configuration** > **Software Settings**. In the right hand pane, double click SecureLogin. The SecureLogin management interface will open up.
8. In the SecureLogin management interface, select the **Preferences** tab. Set each setting in accordance with what you want the users to do.

   NOTE: The users referred in this document are administrators and help desk staff. They have full access to the SecureLogin client. Your configuration might differ slightly.

   The preferences highlighted are the ones that are critical to ensure users are able to mange SecureLogin. Ensure that they are set as shown in the following figure.



Add application prompts for Internet ...	No	<Current GPO .
Add application prompts for Java app...	No	Default
Add application prompts for Windows...	No	<Current GPO .
Allow single sign-on to Internet Explorer	Yes	Default
Allow single sign-on to Java applicati...	Yes	Default
Allow single sign-on to Netscape	No	<Current GPO .
Allow single sign-on to Windows appl...	Yes	Default
Allow user to backup/restore	Yes	Default
Allow users to modify User ID descrip...	Yes	Default
Allow users to view and change Pref...	Yes	Default
Allow users to view and modify API p...	No	<Current GPO .
Allow users to view and modify Appli...	Yes	Default
Allow users to view passwords	No	<Current GPO .
Change the cache refresh interval (in...	60	<Current GPO .
Detect incorrect passwords	No	<Current GPO .
Disable single sign-on	No	Default
Display the system tray icon	Yes	Default
Enable cache file	Yes	Default
Enable the New Login Wizard on the...	Yes	Default
Enforce passphrase use	Yes	<Current GPO .
Password protect the system tray icon	Yes	<Current GPO .
Remove advanced settings	No	Default
Stop walking here	No	Default

9 Click **Ok** on the SecureLogin management interface. This might take a minute to save.

10 Close the GPO editor.

11 In the GPM, select the new GPO you created, remove the Authenticated Users group, and add the admin and help desk groups you created in the previous two sections.

12 Link this policy to the OU where the users are located. Right click and select **Link to an existing GPO**.

13 Select the GPO you created, click OK.

14 Close the GPM. Click **OK** on the group policy tab.

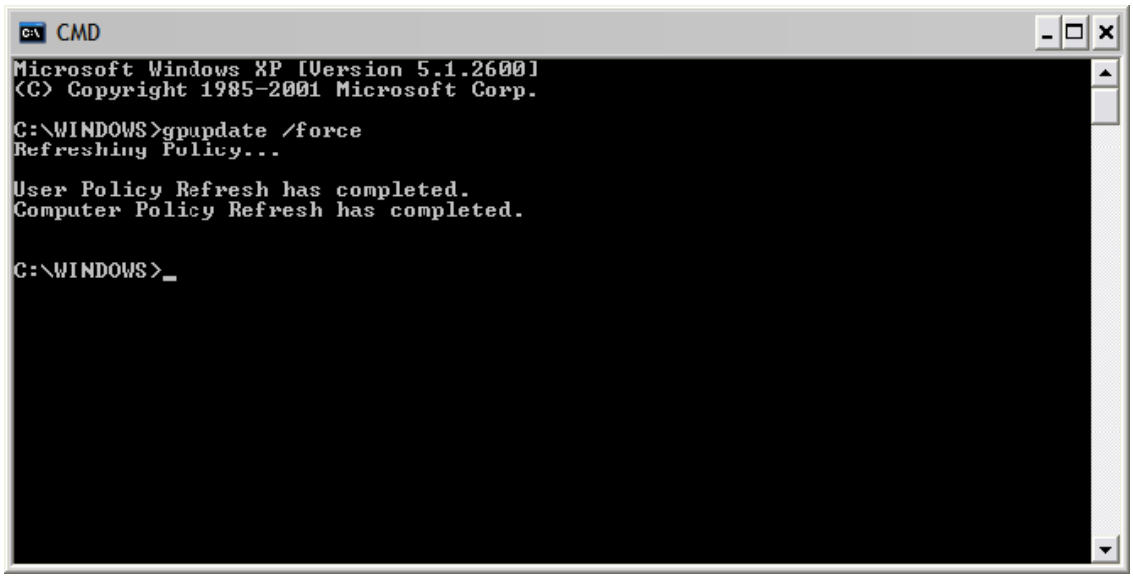
15 Close Active Directory Users and Computers.

Testing your configuration

If you chose to use individual assignment or GPO assignment, proceed with the following tests to confirm your updated configuration

1 On a workstation with SecureLogin and the Active Directory Admin Pack, login as a user who is a member of one of the groups you have configured as SecureLogin administrators or help desk.

2 If your GPO refresh has not occurred, you can manually force the update by going to a command line and issuing the gpupdate /force command (Windows XP). You should see results similar to the following:



3 Launch Active Directory Users and Computers. Navigate to the container where you delegated control. As a member of the Admins group you should be able to manage the OU's, and subordinate objects, applications and preferences.

As a member of the Help Desk group you should be able to only make changes to the users in the OU. It might appear that as a help desk user you can save changes to the OU, but that is not the case. And if you close the Single Sign-On properties and then open it back up, you will see the changes were not saved.

24 Limiting Concurrent Connections

SecureLogin when installed in the LDAP mode (for eDirectory or non-eDirectory), can restrict the number of concurrent user connections to eDirectory, or Active Directory. According to the limit set for a user at the directory level, SecureLogin decides whether to allow user authentication or deny it. The connection log of all the user accounts is maintained and managed by SecureLogin.The validity of any entry will be decided based on the timestamp and the pre-defined time-to-live parameter.

Setting Up the Environment for Limiting Concurrent Connections

Registry Entry

Modify the `EnforceConcurrentConnections` value under `HKLM\Software\Novell\Login\LDAP. This should be a DWORD type` and the value should be set to 1. To disable this feature, set the value to 0. Also add the value of `LDAPContextlessSearchBindcreds` under `HKLM\SOFTWARE\Novell\Protocom\SecureLogin`. This value should be copied from the output file that contains the encrypted credential in the string format. For more information about encrypting the credentials of the LDAP user, refer Contextless Login in the *NetIQ SecureLogin Installation Guide*.

Schema Extension

In order to facilitate this feature we need to extend the schema so that we can add the attributes for the limit of concurrent connections, and the timestamp for a connection.

For eDirectory: The schema has to be extended and the attribute rights have to be set using the included .sch and .ldif files, which are found at `SecureLogin\Tools\Schema\LDAP`. The `Concurrent_schema_extn.sch` file is used to add the attributes to the schema, and the `concurrent-rights.ldif` file is used to extend the rights. These files can be extended as mentioned in the proceeding options.

The `Concurrent_schema_extn.sch` file can be used to extend the eDirectory schema with one of the following options:

- **ndssch (eDirectory schema extension utility):** This is a Windows/ Linux executable. Type the following in the command shell:
  - `ndssch <AdminDN> Concurrent_schema_extn.sch`

  For more information, see ndssch Utility.

- **ICE Tool (version 20503.02 or later):** Execute the following command:
  - `ice -S SCH -f Concurrent_schema_extn.sch -D LDAP -d <AdminDN> -w <password> -L <ServerCertificate>`

    For more information about ICE (NetIQ Import Conversion Export Utility), see NetIQ Import Conversion Export Utility in the *eDirectory Administration* guide.

The `concurrent-rights.ldif` file can be extended by using either of the following options:

- **ICE Tool (version 20503.02 or later):** Execute the following command in eDirectory:
  - `ice -S LDIF -f concurrent-rights.ldif -D LDAP -d <AdminDN> -w <password> -L <ServerCertificate>`

    For more information about ICE (NetIQ Import Conversion Export Utility), see NetIQ Import Conversion Export Utility in the *eDirectory Administration* guide.
- **LDAP Modify tool:** Execute the following command in eDirectory:
  - `ldapmodify -x -h <host ip address> -p 389 -D cn=admin,o=context -w password -f concurrent-rights.ldif`

NOTE: `LDIF` and `SCH` files are not integrated with the `ldapschema.exe` file, but are bundled as separate files in `SecureLogin\Tools\Schema\LDAP`.

For Active Directory: The `ConcurentSchema.exe` file is used to extend the schema to add the required attributes to the schema. The default location for this file is `SecureLogin\Tools\Schema\AD` To extend the schema, perform the following:

1. Run the `Concurrentschema.exe` file.
2. Select **Extend Active Directory Schema**.
3. Assign rights to the directory.

   To set the attribute values refer "Setting the Attribute Values" on page 176
4. Click **OK**.

After the schema is extended, three new attributes are added to the list of attributes:

- **Protocom-SSO-Connections:** This attribute stores the connection information, that is the IP address along with the timestamp. This attribute gets added when a user connects for the first instance.
- **Protocom-SSO-ConnectionLimit:** This attribute stores the configuration parameter, indicating the number of concurrent connections that are allowed for the user.
- **Protocom-SSO-ConnectionTTL:** This attribute stores the configuration parameter that indicates how long the connection information will be stored.

Setting the Attribute Values

To set the attribute values by using iManager or MMC (Microsoft Management console), you should assign the `Protocom-SSO-ConnectionLimit` attribute and the `Protocom-SSO-ConnectionTTL` attribute to the user and then set the attribute values.

To set the attribute values by using iManager, perform the following steps:

1. In the iManager console, click **Roles and Tasks** > **Directory Administration** > **Modify Objec**t.
2. Select the user, then click **OK**.
3. Click **General** > **Other**.

4 Select the attribute from the list under **Unvalued Attributes**, then click ⬅ .

5 In the Add Attribute window, set the attribute value, then click **OK**.

To set the attribute values by using **Microsoft Management Console**, perform the following:

1 In **Active Directory Users and Computers**, click **View > Advanced Features**

2 From the Users list right-click on the required user, then click **Properties > Attribute Editor**.

You can set the attribute values, then click **OK**.

Example:

The attributes are set to the following values:

- Protocom-SSO-ConcurrentConnectionLimit: 2
- Protocom-SSO-ConcurrentConnectionTTL: 1440 (in minutes)

When UserA logs in from workstation 1 with IP 1.1.1.1, a new entry is added to the `Protocom-SSO-Connections` attribute in the IP@timestamp format, that is, 1.1.1.1@20110621000000 (2011 June 21 00 AM).

Similarly, when UserA logs in from workstation 2 with IP 2.2.2.2, another entry is added to the `Protocom-SSO-Connections` attribute in the IP@timestamp format, that is, 2.2.2.2@20110621040000 (2011 June 21 04 AM).

If UserA then tries to log in from workstation 3, SecureLogin will deny the authentication because the connection limit is exceeded.

25 Support for Advanced Authentication

SecureLogin uses the Advanced Authentication infrastructure to integrate SecureLogin features with Advanced Authentication. The integration with Advanced Authentication helps SecureLogin to single sign-on to the required application by using different authentication mechanism. For more information about Advanced Authentication, see the Server Administration Guide.

The authentication mechanisms supported are:

- Biometric authentication using fingerprint
- Smart Card
- Radius client
- Email OTP
- Emergency password
- LDAP password
- OATH OTP, namely HOTP AND TOTP
- Advanced Authentication password
- SMS OTP
- Security questions
- Smartphone
- U2F
- PKI
- Voice

NOTE: NSL 8.5 does not support the NotarisID method.

In a system with Advanced Authentication installed, SecureLogin supports the following features:

- **Authentication:** Advanced Authentication provides the Credential Provider in the system where it is installed. When the user logs in using this Credential Provider, SecureLogin will also launch seamlessly in an Active Directory environment.
- **Re-authentication:** SecureLogin leverages this functionality by making Advanced Authentication available during re-authentication. This module can be configured from the wizard or a script. An appropriate authenticator can be selected for re-authentication when configuring this module. This feature is supported in Active Directory, AD LDS and eDirectory environments.
- **Kiosk Mode:** SecureLogin uses Advanced Authentication as an external authenticator in Active Directory and eDirectory deployments in kiosk mode.
- **DAS Support:** DAS on Active Directory and eDirectory supports the Advanced Authentication and SecureLogin integrated environment. This feature can effectively be used for desktop sharing using DAS.

Advanced Authentication Registry Settings

- **To enable kiosk mode, create the following registry:**

Registry Path	Registry Type	Registry Name	Registry Value
HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin	DWORD	NSLADAUTH	For kiosk authentication, set NSLADAUTH to 1. Setting NSLADAUTH to 0 is the default behavior.

- **To disable advanced authentication, create the following registry:**

Registry Path	Registry Type	Registry Name	Registry Value
HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin	DWORD	NoAuthasas	For disabling the advanced authentication feature, set NoAuthasas to 1; but for enabling it, set to 0.

- **To enable Advanced Authentication debug logging, create the following registry:**

Registry Path	Registry Type	Registry Name	Registry Value
HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\Logging	DWORD	All	For enabling the advanced authentication debug logging, set All to 0. Only log levels 1 and 2 are supported.

- **To disable SSL check, create the following registry:**

Registry Path	Registry Type	Registry Name	Registry Value
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\SecureLogin\AdvancedAuthentication	DWORD	DisableSSLCheck	For disabling the SSL check, set DisableSSLCheck to 1; but for enabling it, set to 0(default).

Configuration of Advanced Authentication for SecureLogin

When installing Advanced Authentication, the default event name is **Windows logon** because SecureLogin kiosk mode with Advanced Authentication recognizes only **Windows logon** as the event configured in the Advanced Authentication server.

You can also enter any custom name (same as configured in the Advanced Authentication server) as the event name, as the custom event can be used for re-authentication.

If you enable the Kiosk mode, you need to enter the username and logon method. The logon method will have the list of chains from the Windows Logon event. For the re-authentication wizard, the list of chains either from the custom or **Windows Logon** event will be displayed for selection. You must choose a logon method to initiate the Advanced Authentication flow.

NOTE: During SecureLogin installation, the installer registers an endpoint to the Advanced Authentication server, which is used for validation of the credentials for all supported mechanisms.

26 Troubleshooting

Unable to See Password Using ?syspassword If the User is Authenticated Using Smart Card PIN

In iManager, execute one of the following steps:

Workaround 1: In **Universal Password > Configuration Options**, select **Allow admin to retrieve passwords**.

Workaround 2: In **Universal Password > Configuration Options**, select **Allow the following to retrieve passwords**. Specify the FDN of the user allowed to retrieve the password.

`java.lang.NullPointerException : null` Error Appears When Accessing the SecureLogin Preferences

After installing the latest SecureLogin single sign-on plug-in(`sso.npm`) in iManager, when you access the SecureLogin preferences `java.lang.NullPointerException : null` error appears.This error occurs because in iManager SecureLogin single sign-on plug-in(`sso.npm`) upgrade did not successfully replace all the old files.

Workaround: In iManager, remove the `jssoapi.jar` or `jsso-api.jar` file from the location `imanager\tomcat\webapps\nps\WEB-INF\lib`, restart the tomcat sever that hosts the iManager and then install SecureLogin single sign-on plug-in(`sso.npm`).

SecureLogin Fails To Provide Single Sign-on to Virtual Applications

Issue: SecureLogin does not provide single sign-on to virtual applications because the virtualization tools do not allow SecureLogin running on the host to access the virtualized application.

For example, when you use ZENworks Application Virtualization to virtualize Internet Explorer, SecureLogin does not provide single sign-on to Internet Explorer. The following reasons can cause this issue:

1. ZENworks Application Virtualization does not allow Internet Explorer to access the existing Internet Explorer registries for the extensions that are configured with SecureLogin.
2. ZENworks Application Virtualization does not allow reading of the COM entries for SecureLogin
3. ZENworks Application Virtualization does not allow reading of the file system where SecureLogin is installed.

Workaround: While virtualizing an application, the virtualization tool must include the registries or the file system sections of the application that SecureLogin will access to provide single sign-on.

For example, if you virtualize Internet Explorer, you must virtualize BHO registry keys also. It will allow SecureLogin to provide single sign-on to Internet Explorer.

SecureLogin Fails To Start For Domain Users On Windows Domains Or Domain Controller Servers

Issue: When domain users that are not assigned with administrator rights starts SecureLogin, SecureLogin displays the `A password change has been detected. For security reasons, it is necessary to logout of windows and log back in.` error message and stops working. This issue occurs on Windows Domains or Domain Controller servers. This issue does not occur on Windows clients.

Workaround: Perform the following steps to fix this issue on a Domain Controller server:

1. Open the **Group Policy Management** console and navigate to **Domain Controllers**.
2. Right click the appropriate Domain Controller Policy and click **Edit**.
3. In the **Group Policy Management Editor** window, click **Computer Configuration**.
4. Navigate to **Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
5. Click **Allow log on locally**.
6. In the **Allow log on locally Properties** window, click **Add User or Group...** to add the appropriate users or group.
7. Click **OK**.

A Error Messages

SecureLogin error messages display a number code that generally includes a text description of the error. SecureLogin error numbers currently range between -101 and -914. Following is a list of these error message, their cause, and the appropriate action to take.

Some of the codes displayed in SecureLogin error messages are not native SecureLogin codes. Refer to the relevant application's Help for assistance with the following:

NetIQ eDirectory: Numbers between -1 and -813

Microsoft Active Directory; Error codes such as, 0x80070002

For more information about Active Directory error codes, go to the Microsoft Web site (http://msdn.microsoft.com)

-102: BROKER_NO_SUCH_ENTRY

Possible Cause: You tried to load an application definition or variable that does not exist.

For example, you set up Terminal Launcher to run from a shortcut or to run a particular application definition, but the application definition does not exist.

Action: Check that the name of the application definition is actually defined in SecureLogin. Verify that the name is the same as the name specified in the application definition.

-103: BROKER_INVALID_CLASS_CREATED

Possible Cause: Data has become corrupted, or you are running an earlier version.

SecureLogin is trying to create a new version of the application definition data format that was stored in ANDS.

Action: Upgrade the older SecureLogin client to the new client. Install the latest SecureLogin software.

-104: BROKER_CREATE_CLASS_FAILED

Possible Cause: The SecureLogin client has run out of memory.

Action: Free up some memory. Try again later.

-105: BROKER_REMOVE_ENTRY_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-106: BROKER_UPDATE_GET_ENTRY_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-107: BROKER_ENTRY_NOT_FOUND

Possible Cause: An attempt to load an application definition or variable that does not exist.

Action: Check that the name of the application definition is actually defined in SecureLogin. Verify that the name is the same as the name specified in the application definition editor.

-109: BROKER_SCRIPT_BUFFER_ALLOC_FAILED

Possible Cause: The SecureLogin client has run out of memory.

Action: Free up some memory. Try again later.

-110: BROKER_NO_MORE_PLATFORMS

Possible Cause: Data is corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-111: BROKER_NO_MORE_VARIABLES

Possible Cause: Data is corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-112: BROKER_NO_SUCH_VARIABLE

Possible Cause: You are trying to use an undefined variable.

Because SecureLogin is not prompting you for the variable, data has become corrupted, or some other situation is preventing the software from working as expected.

Action: Contact NetIQ Support.

-114: BROKER_PRIMARY_NOT_AVAILABLE

Possible Cause: You are not logged on to the directory. You are using the offline cache. Therefore, you cannot perform some directory functions. For example, you cannot change your passphrase.

Action: Log in to the directory.

-116: BROKER_HEADER_DATA_CORRUPT

Possible Cause: Data is corrupted. You might have a customized build for your site, but have installed a standard version of SecureLogin, or have gone from a standard version to a customized build for your site.

Action: Delete the local cache file and try again. If unsuccessful, contact NetIQ Support.

-120: BROKER_INVALID_PREF_DATA_TYPE

Possible Cause: Data is corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-121: BROKER_PREFERENCE_DATA_CORRUPT

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-122: BROKER_TARGET_ENTRY_LIST_NOT_LOADED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-123: BROKER_CACHE_PASSWORD_INCORRECT

Possible Cause: You have tried to log on from offline mode, but the password you entered does not match the expected password from the local cache.

Typically, the offline password is the passphrase answer.

Action: Enter the correct passphrase answer or directory password.

-129: BROKER_ENTRY_LIST_NOT_NULL

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Delete the local cache file and try again. If unsuccessful, contact NetIQ Support.

-130: BROKER_ENTRY_LIST_NULL

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Delete the local cache file and try again. If unsuccessful, contact NetIQ Support.

-131: BROKER_YSM_LIST_NOT_NULL

Possible Cause: Memory is not handled as expected.

Action: Contact NetIQ Support.

-132: BROKER_SYM_LIST_NULL

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-138: BROKER_SYMBOL_DATA_CORRUPT

Possible Cause: Data has become corrupted in the local cache file or in the directory.

Action: Delete the local cache file and try again. If unsuccessful, contact NetIQ Support.

-140: BROKER_SCRIPT_DATA_CORRUPT

Possible Cause: Data has become corrupted in application definitions.

Action: Delete the local cache file and try again.

-141: BROKER_PREF_INVALID

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-142: BROKER_SET_PREF_INVALID

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-145: BROKER_SECURITY_ALERT

Possible Cause: Unable to locate security keys (AuthData), but security data appears to exist. It is possible that someone has attempted to gain access to your security data.

Action: Contact your system administrator.

-166: BROKER_INVALID_DES_KEY

Possible Cause: Hex strings are invalid. The DES_KEY variable requires hexadecimal (0-9, A-F) numbers.

Action: Make sure that the DES_KEY variable contains only hexadecimal numbers.

-167: BROKER_INVALID_DES_OFFSET

Possible Cause: Hex strings are invalid. The DES_OFFSET variable requires hexadecimal (0-9, A-F) numbers.

Action: Make sure the DES_OFFSET variable contains only hexadecimal numbers.

-168: BROKER_DESKEY_NOT_FOUND

Possible Cause: You tried to generate a one-time password for a platform. However, you have not defined the DES_KEY variable.

Action: Create the DES_KEY variable.

-169: BROKER_DESOFFSET_NOT_FOUND

Possible Cause: You tried to generate a one-time password for a platform. However, you have not defined the DES_OFFSET variable.

Action: Create the DES_OFFSET variable.

-171: BROKER_CACHE_FILE_OPEN_FAIL

Possible Cause: SecureLogin tried to read or write to the offline cache. However, SecureLogin is unable to open the cache file.

Action: Assign rights so that the specified user object has rights to the cache directory.

-173: BROKER_NO_MORE_CACHE_FILE_DATA

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-174: BROKER_CACHE_SAVE_FAILED

Possible Cause: SecureLogin is unable to save data to the offline cache.

Action: Assign rights so that the specified user object has rights to the cache directory.

-175: BROKER_CACHE_SECRETS_INCORRECT

Possible Cause: The offline cache password is incorrect for either of the following reasons:

- The key used to decrypt the cache file is not the key that the cache file was encrypted with.
- If you log on as a user to a workstation and create a cache file, and then you go to another workstation, reset your passphrase and log on, then when you return to the original workstation this error message appears.

Action: Delete the cache file.

-176: BROKER_PUBLIC_KEY_READ_FAILED

Possible Cause: SecureLogin is unable to read the public key from Active Directory System.

Action: Troubleshoot Microsoft Active Directory System and Microsoft ADAM.

-177: BROKER_PUBLIC_KEY_HAS_CHANGED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-179: BROKER_RTVALUE_DOES_NOT_EXIST

Possible Cause: You tried to read a runtime variable that is not defined.

Action: Check the application definition. Make sure that the variable is set before it is read or used as a command.

-180: BROKER_DS_VARIABLE_NOT_READ

Possible Cause: You used one of the % variables to read a directory attribute, but SecureLogin cannot read the variable.

Action: Make sure that you have spelled the attribute name correctly. Troubleshoot Microsoft Active Directory System or Microsoft ADAM.

-181: BROKER_WRONG_PASS_PHRASE

Possible Cause: The passphrase or password is incorrect. The reason could be:

- You entered the wrong passphrase.
- You tried to change your passphrase, but entered it incorrectly.
- You password protected the SecureLogin notification area (system tray) icon and entered the incorrect password.

Action: Enter the passphrase or password correctly.

-190: BROKER_NO_AUTH_DATA_FOUND

Possible Cause: Although the SecureLogin Entry attribute has data, the SecureLogin Auth attribute was blank.

Someone deleted the SecureLogin Auth attribute.

Action: Delete the Prot:SSO Entry attribute.

SecureLogin creates these attributes the next time you run SecureLogin.

-192: BROKER_UNABLE_TO_INSTANTIATE

Possible Cause: A module, for example, WinSSO, is unable to connect to the Combroker.

Action: If you are using Windows 95, make sure that you have the latest DCOM update, or reinstall Internet Explorer.

For other platforms, reinstall SecureLogin.

-195: BROKER_FILE_TRAITS_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-196: BROKER_DUMMY_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-199: BROKER_ERROR_COMMAND_NOT_HANDLED

Possible Cause: An application definition parser encountered an unrecognizable command.

Action: Make sure that:

- The command is spelled correctly.
- The If/EndIf blocks match.

-200: BROKER_END_OF_SCRIPT

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-201: BROKER_UNEXPECTED_END_OF_SCRIPT

Possible Cause: `If/EndIf` or `Repeat/EndRepeat` blocks do not match. SecureLogin reached the end of the application definition without finding an expected `EndIf` or `EndRepeat` command.

Action: Check the application definition. Make sure that the `If/EndIg` and `Repeat/EndRepeat` blocks match.

-206: BROKER_BREAK_BLOCK

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-207: BROKER_END_SCRIPT_NOW

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-210: BROKER_CORPORATE_MOD_ABORTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-211: BROKER_ENTRY_ALREADY_ON_LIST

Possible Cause: You tried to add an application definition or variable, but an application definition or variable with that name already exists.

Action: Do one of the following

- Use a different name for the application definition or variable.
- Rename the existing application definition or variable in the application definition editor.

-213: BROKER_NDS_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-214: BROKER_UNABLE_TO_GET_CURRENT_OU

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact NetIQ Support.

-217: BROKER_ARG_NUM

Possible Cause: In application definition language, each command expects a certain number of arguments. You have used either too few or too many arguments for a given command.

Action: Make sure you are passing the correct number of arguments to the command.

-219: BROKER_NOT_A_NUMBER

Possible Cause: The application definition language was expecting a decimal number, but characters other the 0-9 appeared.

Action: Remove incorrect characters.

-220: BROKER_HLLAPI_FUNCTION_NOT_FOUND

Possible Cause: In the Terminal Launcher configuration, you specified a `HLLAPI.DLL` and the name of the function in the DLL. The name of the function cannot be found in the DLL.

Action: Check you have specified the correct terminal emulator type. Make sure that you entered the HLLAPI function correctly.

-221: BROKER_HLLAPI_OBJECT_UNINITIALISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-222: BROKER_HLLAPI_DLL_LOAD_FAILED

Possible Cause: Terminal Launcher was unable to load the `HLLAPI.DLL` that you specified. The HLLAPI.DLL for that emulator is looking for other DLL files that do not exist or are not installed for that emulator.

Action: Make sure that the path and file that you are entered for the DLL are correct.

Check the vendor's documentation for information about that emulator.

-223: BROKER_HLLAPI_OBJECT_ALREADY_INITIALISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-224: BROKER_ERROR_DURING_WINHLLAPICLEANUP

Possible Cause: Terminal Launcher has called the WinHLLAPI cleanup function for a WinHLLAPI emulator.

Action: Check the vendor's documentation for information about that emulator.

-225: BROKER_CANNOT_FIND_WINHLLAPISTARTUP_FUNCTION_IN_DLL

Possible Cause: In the Terminal Launcher configuration, you incorrectly specified that the emulator is a WinHLLAPI emulator.

Action: Make sure that you have specified the correct emulator type.

-226: BROKER_ERROR_DURING_WINHLLAPISTARTUP

Possible Causes: The reason can be the following:

- The terminal emulator does not support the right version of HLLAPI (requires at least V.1.1).
- The attempt to reset a connection to a HLLAPI terminal emulator failed.

Action: Check the vendor's documentation for information about that emulator.

-227: BROKER_CANNOT_FINDWINHLLAPICLEANUP_FUNCTION_IN_DLL

Possible Cause: In the Terminal Launcher configuration, you incorrectly specified that the emulator is a WinHLLAPI emulator.

Action: Make sure you have specified the correct emulator type.

See the NetIQ Web site for information about configuring specific terminal emulators.

-228: BROKER_BUTTON_NOT_FOUND

Possible Cause: For a Windows single sign-on application, no button exists for the control ID you specified. For example, if you specified Click #3, no button exists for control ID #3.

Action: Specify the correct emulator type.

-230: BROKER_SETPLAT_FAILED

Possible Cause: The regular expression that you supplied in the SetPlat command is invalid.

Action: Check the syntax of the regular expression that you provided.

-231: BROKER_AUTH_CANCEL

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-232: BROKER_UNABLE_TO_START_PROGRAM

Possible Cause: The Run command was unable to find and start the requested program.

Action: Make sure that the executable program exists and that the path is correct.

-234: BROKER_FREE_PLATFORM_SCRIPT_NULL_PTR

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-235: BROKER_VBA_LOGIN_INTERFACE_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-236: BROKER_CHANGEPASSWORD_INVALID_VARIABLE_SYNTAX

Possible Cause: One of the parameters that you pass to the ChangePassword command must be a variable. The parameter that you provided is not a variable.

Action: Specify a variable.

-237: BROKER_MAD_COMMAND_SET_INVALID_VARIABLE_SYNTAX

Possible Cause: The first parameter that you pass to the Set command must be a variable. The parameter that you provided is not a variable.

Action: Specify a variable.

-239: BROKER_POLICY_SCRIPT_ARG_NUM

Possible Cause: One of the commands in a password policy script has too few or too many arguments.

Action: Include the correct number of arguments.

-240: BROKER_VALID_CHARS_OUTNUMBERED

Possible Cause: A password is unable to satisfy a password policy. This is because the maximum number of allowable characters is less than the minimum number of allowable characters.

Action: Set the maximum number of a particular class of characters to a greater number than the minimum number of specified allowable characters.

-241: BROKER_PASSWORD_LOGIC_ERROR

Possible Cause: You have incorrectly set up a password policy. No password can satisfy all the settings.

Action: Work through each restriction in the password policy, and make sure that one restriction does not contradict another restriction in the policy.

-242: BROKER_EXECEPTION_CHARACHER_FOUND

Possible Cause: You entered a password that contains a character that is not allowed.

Action: Use allowable characters in your password.

-243: BROKER_PASSWORD_TOO_SHORT

Possible Cause: You entered a password that does not have enough characters.

Action: Provide enough characters in your password.

-244: BROKER_PASSWORD_TOO_LONG

Possible Cause: You entered a password that has too many characters.

Action: Enter the correct number of characters.

-245: BROKER_INSUFFICIENT_UPPERCASE_CHARS

Possible Cause: You entered a password that has too few uppercase characters.

Action: Use the specified number of uppercase characters in your password.

-246: BROKER_TOO_MANY_UPPERCASE_CHARS

Possible Cause: You entered a password that has too many uppercase characters.

Action: Use the specified number of uppercase characters in your password.

-247: BROKER_INSUFFICIENT_LOWERCASE_CHARS

Possible Cause: You entered a password that has too few lowercase characters.

Action: Use the specified number of lowercase characters in your password.

-248: BROKER_TOO_MANY_LOWERCASE_CHARS

Possible Cause: You entered a password that has too many lowercase characters.

Action: Use the specified number of lowercase characters in your password.

-249: BROKER_INSUFFICIENT_PUNCTUATION_CHARS

Possible Cause: You entered a password that has too few punctuation characters.

Action: Use the specified number of punctuation characters in your password.

-250: BROKER_TOO_MANY_PUNCTUATION_CHARS

Possible Cause: You entered a password that has too many punctuation characters.

Action: Use the specified number of punctuation characters in your password.

-251: BROKER_INSUFFICIENT_NUMERALS

Possible Cause: You entered a password that has too few numerals.

Action: Use the specified number of numerals in your password.

-252: BROKER_TOO_MANY_NUMERALS

Possible Cause: You entered a password that has too many numerals.

Action: Use the specified number of numerals in your password.

-253: BROKER_NT_FILE_TRAITS_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-256: BROKER_UNABLE_TO_GET_NT_CHACE_DIR

Possible Cause: You are using Windows NT 4 Domains mode, but you have not defined or mapped a Home drive.

Action: Log in as the user to determine whether the Home drive and Home path variables are set. If the variables are not set, use the Windows NT domain administrative tools to set them.

NOTE: Version 3.6 and above do not support Windows NT.

-257: BROKER_UNABLE_TO_CREATE_NT_CACHE_DIR

Possible Cause: The user object did not have rights to create a directory on the user's local drive.

Action: Grant the user object rights to the directory.

-259: BROKER_MUST_BEGIN_WITH_UPPERCASE

Possible Cause: You entered a password that did not begin with an uppercase character.

Action: Enter an uppercase character at the beginning of the password.

-260: BROKER_NO_DATA_STORES_LOADED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-261: BROKER_ENTRY_SRC_OBJECT_MISMATCH

Possible Cause: You are using a platform other than NDS or eDirectory and have moved an object. The directory object that you are reading entries from is not the directory object that the entries were saved to.

Action: Manually copy and paste the scripts between the objects.

-262: BROKER_CACHE_FILE_INCORRECT_VERSION

Possible Cause: The cache file that you are trying to load was created by a later version of SecureLogin.

Action: Use the version of SecureLogin that created the cache file.

Install the latest version of SecureLogin.

-263: BROKER_DDE_LOGIN_INTERFACE_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-264: BROKER_DDE_CONNECT_FAILED

Possible Cause: Terminal Launcher could not connect to a specified DDE emulator.

Action: Make sure that the emulator launched correctly and the emulator's DDE support is turned on.

-265: BROKER_DDE_DISCONNECT_FAILED

Possible Cause: Failed attempt to disconnect from a DDE-supporting terminal emulator.

Action: See the vendor's documentation.

-266: BROKER_NT_FILE__SAVE_FAILED

Possible Cause: The user object was unable to save to the equivalent of a cache file in the Home directory using Windows NT 4 Domains.

Action: Grant the user object rights so that the user can write files to the Home directory.

NOTE: Version 3.6 and above do not support Windows NT.

-269: BROKER_NOT_A_PASSWORD_POLICY_COMMMAND

Possible Cause: An invalid command was used in a password policy.

Action: Make sure that the command is spelled correctly.

-271: BROKER_PASSWORD_UNACCEPTABLE

Possible Cause: The password did not meet the requirements as specified in password policies.

Action: Enter the password correctly.

-273: BROKER_MSTELNET_OPERATION_NOT_SUPPORTED

Possible Cause: The generic emulator cannot support a particular operation, for example, SetCursor.

Action: Do not use the command for generic emulators.

-279: BROKER_EMULATOR_LAUNCH_FAILED

Possible Cause: In Terminal Launcher, you can configure the path to the executable that will run. However, the specified executable is unable to run.

Action: Make sure the path to the emulator is correct.

-280: BROKER_UNABLE_TO_CREATE_EMULATOR

Possible Cause: You have specified an invalid terminal type in TLAUNCH.INI (or the Terminal Launcher configuration).

Action: Specify the correct terminal type.

-281: BROKER_INVALID_CHARACTER_FOUND_IN_PASTE_ID_LIST

Possible Cause: A comma does not separate decimal numbers for copy IDs.

Action: For generic emulators, you must specify a set of copy control IDs. Use a comma to separate decimal numbers.

-282: BROKER_INVALID_CHARACTER_FOUND_IN_COPY_ID_LIST

Possible Cause: A comma does not separate decimal numbers for copy IDs.

Action: For generic emulators, you must specify a set of copy control IDs. Use a comma to separate decimal numbers.

-283: BROKER_UNABLE_TO_READ_TLAUCH_INI

Possible Cause: SecureLogin is unable to read the `TLAUNCH.INI` file because the file has been deleted.

Action: Do one of the following:

- Create a blank `TLAUNCH.INI` file.
- Return to the default `TLAUNCH.INI` file by reinstalling SecureLogin.

-284: BROKER_NO_TERMINAL_TYPE_DEFINED

Possible Cause: The `TLAUNCH.INI` file contains an error. The terminal type for the emulator is not defined.

Action: Use Terminal Launcher to specify a terminal type for the emulator.

-285: BROKER_EMULATOR_INFO_NOT_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-286: BROKER_RELOAD_NOT_ENABLED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-287: BROKER_TERMINAL-CONNECT-TRY-AGAIN

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-289: BROKER_WRONG_OBJECT_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-290: BROKER_FILE_LOAD_FAILED

Possible Cause: You do not have enough rights to convert an earlier `TLAUNCH.INI` file to a later format.

Action: Do one of the following:

- Read an earlier `TLAUNCH.INI` file.
- Create a new `TLAUNCH.INI` file.

NOTE: Ask the administrator to assign you necessary rights.

-292: BROKER_DLL_NOT_INITIALISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-294: BROKER_SETPLAT_VARIABLE_MUST_BE_RUN_TIME

Possible Cause: The first argument to a `SetPlat` argument can be a variable. The variable used is not a runtime variable.

Action: Make the first argument a runtime variable.

-295: BROKER_ERROR_CONDITIONAL_COMMAND_NOT_HANDLED

Possible Cause: SecureLogin does not handle text in the second part of an If command.

Action: Make sure that the command is the one listed and documented correctly.

-297: BROKER_PARSER_ELSE_STATEMENT_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-298: BROKER_RAW_MODE_MUST_BE_SECOND_ARG

Possible Cause: For the Click command, you have placed the –X and –Y arguments before –Raw.

Action: If you use –Raw, place it as the first argument.

-299: BROKER_DISALLOWED_REPEATS_EXIT

Possible Cause: You have tried to use repeated characters in a Password Policy that does not allow them.

Action: Avoid repeated characters.

-300: BROKER_DISALLOWED_SEQUENTIALS_EXIST

Possible Cause: You have tried to use sequential characters in a password, but a Password policy does not allow them.

Action: Avoid sequential characters.

-301: BROKER_DISALLOWED_KEYBOARD_ADJACENTS_EXIST

Possible Cause: You entered a password that has an unacceptable sequence of characters.

Action: Enter an approved sequence of characters.

-303: BROKER_CHARACTER-NOT-IN-REQUIRED-POSITION

Possible Cause: You entered a password that does not have a character in a required position.

Action: Enter the password correctly.

-308: BROKER_BAD_POSITION_ARGUMENT

Possible Cause: While calling a SetCursor command, you tried to move the cursor to an invalid position. For example, out of the terminal session's boundary.

Action: Specify a valid position.

-309: BROKER_ERROR_CONVERTING_POSITION

Possible Cause: The conversion from –X and –Y coordinates for the SetCursor command has failed.

Action: Specify the –X and –Y coordinates for one offset from the top left-hand corner of the screen.

-310: BROKER_NOT_A_WRITABLE_VARIABLE

Possible Cause: You tried to save a new value to a type of variable that cannot be written to.

Action: Use a runtime or normal variable.

-311: BROKER_RUN_SCRIPT_AGAIN

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-312: BROKER_NO_OU_PERIOD_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-314: BROKER_COPY_BACKUP_FAILED

Possible Cause: When SecureLogin begins to update the cache file, SecureLogin first copies the current cache file to a file with the same name, but uses the extension `.GOOD`.

SecureLogin was unable to copy the file. The `.GOOD` file is already open because another process is using it.

Possible Cause: You do not have rights to create files in the directory.

Action: Ask the administrator to assign you rights to the directory.

-315: BROKER_GOTO_LABEL_ALREADY_DEFINED

Possible Cause: You have used a `GoTo` command, but the label that you directed it to has already been used.

Action: Remove the second label command.

-316: BROKER_GOTO_LABEL_NOT_DEFINED

Possible Cause: You have used a `GoTo` command, but the label that you directed it to has not been defined.

Action: Define the label.

-317: BROKER_INCORRECT_DATABASE_VERSION

Possible Cause: The version of SecureLogin that you are using does not handle the version of SecureLogin that is stored in the directory.

Action: Upgrade to the latest version of SecureLogin.

-318: BROKER_DIRECTORY_CRC_DOES_NOT_MATCH

Possible Cause: Whenever SecureLogin stores an entry in Microsoft Active Directory, SecureLogin employs a redundancy check. If the redundancy check does not match when SecureLogin reloads the entry, then the data in Microsoft Active Directory has been corrupted.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-319: BROKER_DISALLOWED_DUPLICATE_EXIST

Possible Cause: You entered a password that has unacceptable duplicate characters.

Action: Enter the password correctly.

-320: BROKER_GOTO_LIST_ASSERTION

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-321: BROKER_SUBROUTINE_NOT_DEFINED

Possible Cause: A Call command is calling a subroutine that has not yet been defined.

Action: Define the subroutine.

-322: BROKER_UNABLE_TO_FIND_PASSWORD_FIELD

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-323: BROKER_PASSWORD_FIELD_STYLE_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-324: BROKER_WEB_ACTION_NOT_SUPPORTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-325: BROKER_ENTRY_MUST_HAVE_NON_NULL_KEY

Possible Cause: You tried to add an application definition or variable that is a blank string.

Action: Provide a name for the application definition or variable.

-326: BROKER_VARIABLE_REQUIRED

Possible Cause: Some commands, for example, ReadText, require a variable to copy the data that they are returning to. The argument must be a variable.

Action: Change the argument to a variable.

-327: BROKER_OBJECT_NOT_FOUND

Possible Cause: Microsoft Active Directory/ADAM library was unable to allocate memory.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-328: BROKER_ADS_MEMORY_FAILURE

Possible Cause: The Microsoft Active Directory/ADAM library was unable to allocate memory.

Action: Close one or more applications and try again.

-329: BROKER_ADS_ERROR_GETTING_ATTRIBUTE

Possible Cause: Although data exists in Microsoft Active Directory/ADAM, SecureLogin is unable to read the data.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-330: BROKER_ADS_INSUFFICIENT_RIGHTS_TO_DELETE

Possible Cause: When you removed an application definition, SecureLogin tried to delete part of an attribute from Microsoft Active Directory/ADAM. However, you are unable to delete the attribute because you do not have sufficient rights.

Action: The administrator must assign sufficient directory rights for each user object so that the user can modify SecureLogin attributes.

-331: BROKER_ADS_ERROR_DELETING_VALUE

Possible Cause: Microsoft Active Directory/ADAM was unable to delete a value.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-332: BROKER_NO_PASSWORD_FIELD_VARIABLE_IN_SCRIPT

Possible Cause: A Web application definition must have at least one Type command that has “password” as the second argument.

The following lines illustrate a typical application definition:

- `Type $Username`
- `Type $Password Password`

However, the application definition has no `Type` command followed by the Password attribute.

Action: Add a Type command followed by the Password attribute.

-333: BROKER_REGEX_GET_REPLACE_STRING_FAILED

Possible Cause: On the RegSplit command, the string that you are running through the regular expression did not match.

Action: Change the regular expression.

-335: BROKER_REGEX_COMPILE_FAILED

Possible Cause: The syntax of the regular expression was incorrect.

Action: Revise the syntax of the regular expression.

-336: BROKER_DIRECTORY_AUTH_DATA_CORRUPT

Possible Cause: The SecureLogin:SSOAuth data attribute has become corrupt.

Action: Contact NetIQ Support.

-337: BROKER_DES_KEY_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-338: BROKER_DES_INVALID_BLOCK_LEN

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-339: BROKER_INVALID_ENCRYPTION_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-340: BROKER_UNKNOWN_DATABASE_VERSION

Possible Cause: You are using an earlier version of SecureLogin.

Action: Upgrade to the latest version of SecureLogin.

-341: BROKER_USER_KEY_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-343: BROKER_PRIMARY_KEY-DECRYPT_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-344: BROKER_SECONDARY_KEY_DECRYPT_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-345: BROKER_MERGE_WRONG_ENTRY_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-348: BROKER_PASSWORD_RESET_DETECTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-349: BROKER_UNABLE_TO_FIND_SESSION_FILE

Possible Cause: Terminal Launcher could not find a session file for an emulator.

Action: Configure Terminal Launcher with the correct path to the file for the emulator session.

-352: BROKER_AUTH_DATA_INCORRECT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-353: BROKER_RECURSIVE_SCRIPT_INCLUDE_DETECTED

Possible Cause: While using the Include command, you included an application definition twice.

Action: Only include an application definition once.

-354: BROKER_NETWORK_PASSWORD_INCORRECT

Possible Cause: You have turned on the option to prompt the user for the network password before the user can access options on the taskbar, and the user entered an incorrect password.

Action: Enter the correct password.

-355: BROKER_USER_ABORTED_LOAD_PROCESS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-356: BROKER_INVALID_CHARACTER_FOUND_IN_STARTUP_ID_LIST

Possible Cause: For generic emulators, you specify the startup control ID.

A comma must separate a list of numbers. You have used a character other than a comma.

Action: Remove unacceptable characters.

-357: BROKER_ERRO_REG_CACHE_NO_DETAILS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-358: BROKER_ERROR_REG_CHACE_SAVE_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-359: BROKER_ERROR_REG_CACHE_SPLIT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-360: BROKER_PASSWORD_VARIABLE_NOT_ALLOWED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-361: BROKER_NMAS_DLL_NOT_AVAILABLE

Possible Cause: SecureLogin cannot pad the DLL file for NMAS for use with the AAVerify command.

Action: To use features for AAVerify, install NMAS.

-362: BROKER_NMAS_LEGACY_RELOGIN_NOT_FOUND

Possible Cause: SecureLogin could not find the NMAS relogin function in the DLL for NMAS.

Action: Install the latest version of NMAS.

-363: BROKER_STANDARD_VARIABLE_REQURED

Possible Cause: A ? variable has been used and this command requires a $ variable.

Action: Provide a $ variable.

-364: BROKER_LDAP_LOGIN_CANCELLED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-365: BROKER_LDAP_INIT_FAILED

Possible Cause: The initialization of the LDAP SSL layer failed.

Action: Contact NetIQ Support.

-367: BROKER_REG_AUTH_CACHE_MISMATCH

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-368: BROKER_LDAP_TOKEN_DELETED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-369: BROKER_CRED_LIST_NOT_NULL

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-370: BROKER_CRED_LIST_NULL

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-371: BROKER_NO_MORE_CRED_SETS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-372: BROKER_ACCESS_IS_DENIED

Possible Cause: For LDAP, you do not have rights to the area of the directory that you are trying to access.

Action: Grant user objects the correct rights.

-373: BROKER_HLLAPI-CONNECT_FAILED

Possible Cause: Terminal Launcher was unable to connect to the emulator.

Action: Make sure that the emulator has HLLAPI enabled.

-374: BROKER_DUPLICATE_ENTRIES_EXIST

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-375: BROKER_NOT_RUNNING_NT

Possible Cause: Although you are not running Windows NT, you tried to use a feature that is available only through Windows NT.

Action: Do not use that feature unless you are running Windows NT.

-376: BROKER_WINNT_CACHE_AUTH_REG_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-377: BROKER_WINNT_CACHE_AUTH_REG_WRONG_USER

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-378: BROKER_INVALID_PIPE_STRING_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-379: BROKER_HEX_LENGTH_INCORRECT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-380: BROKER_HLLAPI_NOT_CONNECTED_TO_PS

Possible Cause: Terminal Launcher tried to use a HLLAPI function. However, the HLLAPI DLL is not connected to the emulator presentation space.

Action: Make sure that Terminal Launcher is set up correctly with the emulator.

-381: BROKER_HLLAPI_SPECIFYING_PARAMETERS_ERROR

Possible Cause: Incorrect parameters were given to a command that uses a HLLAPI function.

Action: Contact NetIQ Support.

-382: BROKER_HLLAPI_INVALID_PS_POSITION

Possible Cause: An attempt was made to move the cursor or read text from an invalid (out of bounds) position on the emulator presentation space.

Action: Correct the positioning parameter in the application definition.

-383: BROKER_HLLAPI_SYSTEM_ERROR

Possible Cause: Terminal Launcher is not configured correctly for the emulator.

Action: Make sure that Terminal Launcher is set up correctly with the emulator and that the emulator correctly supports HLLAPI.

-384: BROKER_HLLAPI_PS_BUSY_ERROR

Possible Cause: A HLLAPI function is being called while the emulator presentation space is unavailable.

Action: Make sure that the emulator is not being used by another HLLAPI application.

-385: BROKER_HLLAPI_INPUT_REJECTED

Possible Cause: The emulator rejected an attempt to input data into the emulator presentation space.

Action: Make sure that the emulator presentation space is not locked.

-386: BROKER_HLLAPI_ERROR_QUERYING_SESSIONS

Possible Cause: SecureLogin is unable to query available HLLAPI sessions.

Action: Make sure that the Terminal Launcher is set up correctly with the emulator.

-387: BROKER_LAST_NDS_USER_NOT_FOUND

Possible Cause: The last NDS or eDirectory user object, as stored in the registry, could not be read for use in an NMAS login.

Action: Make sure the last NDS or eDirectory user object is stored correctly in the registry.

-388: BROKER_LAST_NDS_USER_UNWORTHY

Possible Cause: The last NDS or eDirectory user object, as stored in the registry, was not in the correct format. An NMAS login was unable to use the format.

Action: Make sure the last NDS or eDirectory user object is stored correctly in the registry.

-389: BROKER_NMAS_DISCONNECTED_LOGIN_NOT_FOUND

Possible Cause: NMAS disconnected login function not found in `NMAS.DLL`.

Action: Make sure that the correct `NMAS.DLL` is installed.

-390: BROKER_LDAP_SSL_INIT_FAILED

Possible Cause: SecureLogin could not initialize the LDAP SSL libraries.

Action: Contact NetIQ Support.

-391: BROKER_LDAP_SSL_ADD_CERT_FAILED

Possible Cause: SecureLogin could not open the certificate you supplied for LDAP over SSL. Either the file does not exist or it is in the incorrect format. If the certificate file specified ends in `.DER`, then SecureLogin uses Distinguished Encoding Rule (DER) format. Otherwise SecureLogin uses B64 format.

Action: Make sure that the path to the certificate is correct and that it is in DER format.

-392: BROKER_BUILTIN_VARIABLE_NOT_FOUND

Possible Cause: A built-in variable such as `?sysversion` was not found.

Action: Check that the format of the variable name is `?SysVersion(system)`.

-393: BROKER_SCRIPT_NOT_PURELY_INDEXED

Possible Cause: While working with Web modules, you mix indexed and nonindexed commands.

For example, you entered the following:

```
Type $Username #1
Type $Password
```

Action: Make sure that all commands use indexes, or remove all indexes.

-394: BROKER_LDAP_PASSWORD_INCORRECT

Possible Cause: The password supplied to log in to LDAP was incorrect.

Action: Check the password.

-395: BROKER_LDAP_USER_NON_EXISTANT

Possible Cause: The user name that you used to log on to LDAP does not exist.

Action: Make sure that the user name exists in the directory and that the LDAP context is correct.

-396: BROKER_LDAP_SERVER_DETAILS_INCORRECT

Possible Cause: One or more of the LDAP server parameters supplied was incorrect.

Action: Check the LDAP server address and port number.

Make sure that the LDAP server you are connected to is running.

-398: BROKER_WIZ_CP_WRONG_SCRIPT_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-399: BROKER_DIVIDE_BY_ZERO_IS_BAD

Possible Cause: Using the Divide command, you attempted division by zero.

Action: Do not attempt division by zero.

-400: BROKER_WRONG_SECTION_NAME

Possible Cause: You manually edited a wizard-generated application definition.

Action: When editing an application definition, do not edit the specially generated comments. Only edit the actual commands. If this error occurs, you will no longer be able to use the wizard for that application definition.

-401: BROKER_INVALID_GLOBAL_WIZARD_CONFIG

Possible Cause: You manually edited a wizard-generated application definition.

Action: Do not edit the specially generated comments in an application definition. Only edit the actual commands. If this error occurs, you will no longer be able to use the wizard for that application definition.

-402: BROKER_LDAP_ATTRIBUTE_DOES_NOT_EXIST_IN_SCHEMA

Possible Cause: Either of the following:

- You are running LDAP on eDirectory, but have not correctly mapped the LDAP attributes.
- You are running LDAP on a platform other than eDirectory. However, the schema is not extended for that platform.

Action: Check your LDAP attribute mappings. Extend the LDAP schema.

-403: BROKER_AAVERIFY_DLL_NOT_AVAILABLE

Possible Cause: SecureLogin was unable to load `SL_AAVERIFY.DLL`.

Action: Make sure that you have the correct DLLs installed for `AAVERIFY`.

-404: BROKER_AAVERIFY_FUNCTION_NOT_FOUND

Possible Cause: You are using the incorrect version of `SL_AAVERIFY.DLL`.

Action: Check the version of `SL_AAVERIFY.DLL`.

-405: BROKER_AAVERIFY_CONSISTENCY_FAILURE

Possible Cause: You are using the incorrect version of `SL_AAVERIFY.DLL`.

Action: Check the version of `SL_AAVERIFY.DLL`.

-406: BROKER_AAVERIFY_ERROR

Possible Cause: You are using the incorrect version of `SL_AAVERIFY.DLL`.

Action: Check the version of `SL_AAVERIFY.DLL`.

-408: BROKER_DES_KEY_DATA_CORRUPT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-409: BROKER_OPERATION_ABORTED_BY_USER

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-410: BROKER_NOT_A_STRING_ATTRIBUTE

Possible Cause: You are using % variables, but the attribute you are reading is not a plain string attribute (SYN_CE_STRING or SYN_CI_STRING on eDirectory).

Action: Check the schema definition of the attribute to confirm that the syntax is SYN_CE_STRING or SYN_CI_STRING.

-411: BROKER_LDAP_INVALID_DN_SYNTAX

Possible Cause: The format of your LDAP user name was invalid.

Action: Check the format of the user name that you entered.

-412: BROKER_INVALID_OPTION_COMBINATION

Possible Cause: An invalid combination of options was passed to an application definition command.

For example, you passed –Right and –Raw to the `Click` command.

Action: See the appropriate application definition command.

-413: BROKER_AAVERIFY_SLOGIN_DOES_NOT_EXIST

Possible Cause: `SL_AAVERIFY.DLL` generates these errors. There is a problem connecting to the SecureLogin server.

Action: Troubleshoot service location problems by reviewing documentation on SecureLogin Advanced Authentication.

-414: BROKER_AAVERIFY_ERR_SLOGIN_NOT_RUNNING

Possible Cause: `SL_AAVERIFY.DLL` generates these errors. There is a problem connecting to the SecureLogin server.

Action: Troubleshoot service location problems by reviewing documentation on SecureLogin Advanced Authentication.

-415: BROKER_AAVERIFY_ERR_LOAD_LIB_SLPAM

Possible Cause: `SL_AAVERIFY.DLL` generates these errors. There is a problem connecting to the SecureLogin server.

Action: Troubleshoot service location problems by reviewing documentation on SecureLogin Advanced Authentication.

-416: BROKER_WI_GETEXENAME_ERR

Possible Cause: The wizard was unable to retrieve the executable name for the window you selected.

Action: Do not use the wizard for this application.

-417: BROKER_ADS_PUT_OCTET_INSUFFICIENT_RIGHTS

Possible Cause: You do not have sufficient rights to Microsoft Active Directory/ADAM to perform the current operation.

Action: Ask the directory administrator to assign you additional Microsoft Active Directory/ADAM system rights.

-418: BROKER_ADS_CLR_OCTET_INSUFFICIENT_RIGHTS

Possible Cause: You do not have sufficient rights to Microsoft Active Directory/ADAM to perform the current operation.

Action: Ask the directory administrator to assign you additional Microsoft Active Directory/ADAM system rights.

-420: BROKER-_SLAASSO_ERR_CRYPTO_KEY_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-421: BROKER_SLASSO_ERR_UNKNOWN_DATA

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-422: BROKER_SLASSO_OUT_OF_MEMORY

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-423: BROKER_ERROR_INITIALISING_DATA_STORES

Possible Cause: SecureLogin was unable to initialize either the primary or secondary datastore.

Action: Contact NetIQ Support.

-424: BROKER_UNABLE_TO_LOAD_SLOTP_DLL

Possible Cause: `SL0TP.DLL` could not be loaded. This DLL is required for synchronizing one-time password to LDAP directories.

Action: Review documentation for one-time passwords.

-425: BROKER_LDAP_NO_SUCH_ATTRIBUTE

Possible Cause: You have used a % variable on LDAP. However, the requested attribute does not exist.

Action: Check the spelling of the attribute name against the LDAP schema.

-426: BROKER_SYS_VARIABLE_NOT_AVAILABLE

Possible Cause: A system variable, for example, ?syspassword, was requested but was not available. `SLINA.DLL`, `SLNMAS.DLL or SLCREDMAN.DLL` must be correctly installed for these variables to function.

Action: Make sure that either `SLINA.DLL`, `SLNMAS.DLL` or `SLCREDMAN.DLL` is installed.

-427: BROKER_USERNAME_UNSUITABLE_FOR_READING_SLINA_CREDS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-428: BROKER_NO_EXCEPTION_HANDLER_DEFINED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-429: BROKER_EXCEPTOPN_RAISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact NetIQ Support.

-430: BROKER_MUST_BE_CALL_OR_GOTO

Possible Cause: When using the `OnException` command, the second parameter must be Call or `GoTo`.

Action: Make sure that Call is used with the `OnException` command.

-442: BROKER_CHAR_UCASE_NOT_IN_REQUIRED_POSITION

Possible Cause: There is not an uppercase character in a position where one is required.

Action: Check the password for compliance with the Password Policy.

-443: BROKER_CHAR_LCASE_NOT_IN_REQURED_POSITION

Possible Cause: Raised by the Password Policy code if there is not a lower case character in a position where one is required.

Action: Check the password for compliance with Password Policy.

-444: BROKER_PUNCTUATION_NOT_IN_REQUIRED_POSITION

Possible Cause: There is not a punctuation character in a position where one is required.

Action: Check password for compliance with Password Policy.

-477: BROKER_UNABLE_TO_GET_REGISTRY_DATA

Possible Cause: The SecureLogin application definition `GetReg` command could not read the required registry information.

Action: Contact NetIQ Support.

-478: BROKER_ERROR_PARSING_PARAMETER

Possible Cause: The registry entry name passed to the SecureLogin application definition `GetReg` command was incorrect.

Action: Make sure that the name begins {HKCR, HKCC, HKCU, HKLM, or HKU} and corresponds to one of the Windows registry hives. Also, it must contain the path to the desired registry entry within the node.

-481: BROKER_AUTH_QUERY_ON_WRONG_OBJECT_TYPE

Possible Cause: SecureLogin has attempted to load data from a directory object of an incorrect type.

Action: Contact NetIQ Support.

-482: BROKER_VERSION_NO_ROLL_BACK

Possible Cause: The SecureLogin datastore version cannot be returned to an older datastore version after it has been set to version 6.0.

Action: Contact NetIQ Support.

-483: BROKER_SECURE_CONNECTION_REQUIRED

Possible Cause: SecureLogin cannot load sensitive data from the server over insecure connections.

Action: Contact NetIQ Support.

-500: BROKER_ERROR-ACCOUNT-EXPIRED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your user account has expired.

Action: Contact your system administrator.

-501: BROKER_ERROR_ACCOUNT_DISABLED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your user account has been disabled.

Action: Contact your system administrator.

-502: BROKER_ERROR_ACCOUNT_LOCKED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your user account has been locked.

Action: Contact your system administrator.

-503: BROKER_ERROR_PASSWORD_EXPIRED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your password has expired.

Action: Change your Active Directory password or contact your system administrator.

-600: BROKER_NONFIR_INVALID_TARGET

Possible Cause: A non-directory datastore is unable to load the local rule that contains the required data for an object. This could be because of the following:

- Insufficient user permissions.
- File failed to download.
- File has been deleted.

Action: Contact your system administrator.

-2147016656: Error opening specified object

Possible Cause: Microsoft Active Directory code error message (value 0x80072031): There is no such object on the server.

Action: You have entered an incorrect object or container definition when assigning user rights. Reenter the correct object or container definition.

B Schema Updates

This section provides information on the following:

Schema Attributes

SecureLogin adds six schema attributes to the directory. The attributes are added during installation using the appropriate schema extension tool, depending on your choice of directory for SecureLogin data storage. In Active Directory and Active Directory with LDAP environments, `adsschema.exe` is used. For Novell NDS or eDirectory environments, `ldapschema.exe` and `ndsschema.exe` is used.

These attributes are required for the encryption and storage of SecureLogin data against directory objects such as user objects and organizational units. The following descriptions include the type of data stored for each attribute and the security rights required to permit the data to be saved for the SecureLogin client.

Before installing SecureLogin, you need to extend the directory schema. For information on extending the schema, see “Extending the eDirectory Schema”in the *NetIQ SecureLogin Installation Guide*.

The schema tool should run when upgrading to a newer version of SecureLogin to ensure that not only the schema is extended properly but also the appropriate rights are set for objects accessed by SecureLogin.

Active Directory Environments

In Active Directory environments, `adsschema.exe` is used.

Protocom-SSO-Auth-Data

This attribute contains all user-specific authentication data, such as the passphrase.

Attribute Name	Protocom-SSO-Auth-Data

Classes assigned to	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.2

Protocom-SSO-Entries

This attribute contains the following:

- All the user's login credentials, including passwords.
- Specific preferences and application definitions at the user object.
- Corporate application definitions and preferences at the container and organizational unit objects.

Attribute Name	Protocom-SSO-Entries
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.1

Protocom-SSO-Entries-Checksum

This attribute stores a checksum so that the single sign-on client can easily determine whether a complete reload of single sign-on adapter information is required.

Attribute Name	Protocom-SSO-Entries Checksum
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.5

Protocom-SSO-Profile

This attribute stores the address of the organizational unit to be redirected to.

Attribute Name	Protocom-SSO-Profile
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Distinguished Name
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.7

Protocom-SSO-Security-Prefs

This attribute stores the data required for advanced passphrase policies, including administrator set passphrase questions and passphrase help information and settings.

Attribute Name	Protocom-SSO-Security-Prefs
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.3

Protocom-SSO-Security-Prefs-Checksum

A checksum used to optimize reading of the Security Preference attribute.

Attribute Name	Protocom-SSO-Security-Prefs-Checksum
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.6

LDAP Environments

In LDAP environments, `ldapschema.exe` is used.

- “Protocom-SSO-Auth-Data” on page 220
- “Protocom-SSO-Entries” on page 220

Protocom-SSO-Auth-Data

This attribute contains all user-specific authentication data, such as the passphrase.

Attribute Name	Protocom-SSO-Auth-Data
Classes assigned to	User
OID	2.16.840.1.113719.2.26.4.1.1

Protocom-SSO-Entries

This attribute contains the following:

- All the user's login credentials, including passwords.
- Specific preferences and application definitions at the user object.
- Corporate application definitions and preferences at the container and organizational unit objects.

Attribute Name	Protocom-SSO-Entries
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.2.1

Protocom-SSO-Entries-Checksum

This attribute stores a checksum so that the single sign-on client can easily determine whether a complete reload of single sign-on adapter information is required.

Attribute Name	Protocom-SSO-Entries Checksum
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.5.1

Protocom-SSO-Profile

This attribute stores the address of the organizational unit to be redirected to.

Attribute Name	Protocom-SSO-Profile
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.17.1

Protocom-SSO-Security-Prefs

This attribute stores the data required for advanced passphrase policies including administrator set passphrase questions and passphrase help information and settings.

Attribute Name	Protocom-SSO-Security-Prefs
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.4.1

Protocom-SSO-Security-Prefs-Checksum

A checksum used to optimize reading of the Security Preference attribute.

Attribute Name	Protocom-SSO-Security-Prefs-Checksum
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.6.1

Protocom-SSO-Connections

This attribute stores the connection information, ie., the ip address along with the timestamp.

Attribute Name	Protocom-SSO-Connections
Classes assigned to	User
OID	2.16.840.1.113719.2.26.4.7.1

Protocom-SSO-ConnectionLimit

This attribute stores the configuration parameter indicating the number of concurrent connections that are allowed for the user.

- The value of this parameter can be set between 0 and 32.
- The default value is 0, where 0 indicates that unlimited connections are allowed and the feature is disabled for the user.

Attribute Name	Protocom-SSO-ConnectionLimit
Classes assigned to	User
OID	2.16.840.1.113719.2.26.4.7.2

Protocom-SSO-ConnectionTimeToLive

This attribute stores the configuration parameter that indicates how long the connection information will be stored. The value is stored in minutes.

- The value of this parameter can be set between 0 and 65536 (Slightly more than 45 days).
- The default value is 65536. This indicates that any entry in the `Protocom-SSO-Connections` attribute that is older than 45 days is considered outdated and hence will be removed.

Attribute Name	Protocom-SSO-ConnectionTimeToLive
Classes assigned to	User
OID	2.16.840.1.113719.2.26.4.7.3

Security Rights Assignments

This section contains information on the following:

- “User-Based Attributes” on page 222
- “Container-Based Attributes” on page 223

User-Based Attributes

The directory user objects for people using the SecureLogin requires the following attribute rights against their own objects.

Attribute Name	Entry Rights Required
Protocom-SSO-Auth-Data	Read/Write
Protocom-SSO-Entries	Read/Write
Protocom-SSO-Entries-Checksum	Read/Write
Protocom-SSO-Profile	Read/Write
Protocom-SSO-Security-Prefs	Read/Write

Attribute Name	Entry Rights Required
Protocom-SSO-Security-Prefs-Checksum	Read/Write
Protocom-SSO-Connections	Write/Public Read
Protocom-SSO-ConcurrentConnectionLimit	Public Read
Protocom-SSO-ConcurrentConnectionTimeToLive	Public Read

Container-Based Attributes

In addition, users require the following directory attribute rights against all container objects.

Attribute Name	Entry Rights Required
Protocom-SSO-Entries	Read
Protocom-SSO-Entries-Checksum	Read
Protocom-SSO-Profile	Read
Protocom-SSO-Security-Prefs	Read
Protocom-SSO-Security-Prefs-Checksum	Read