

Novell SecureLogin

3.0

www.novell.com

ADMINISTRATION GUIDE

October 31, 2003



Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Copyright © 2002 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

SecureLogin Administration Guide
[October 31, 2003](#)

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

ConsoleOne is a registered trademark of Novell, Inc. in the United States and other countries.

DeFrame is a trademark of Novell, Inc.

DirXML is a registered trademark of Novell, Inc. in the United States and other countries.

eDirectory is a trademark of Novell, Inc.

GroupWise is a registered trademark of Novell, Inc. in the United States and other countries.

NDS is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Client is a trademark of Novell, Inc.

Novell Modular Authentication Services (NMAS) is a trademark of Novell, Inc.

SecretStore is a registered trademark of Novell, Inc.

ZENworks is a registered trademark of Novell, Inc.

ZENworks OnDemand Services is a trademark of Novell, Inc.

Third-Party Trademarks

All other third-party trademarks are the property of their respective owners.

About This Guide

This guide for network administrators provides information on the following:

- ◆ Chapter 1, “Overview: Single Sign-On,” on page 7
- ◆ Chapter 2, “Installing SecureLogin,” on page 19
- ◆ Chapter 3, “Migrating from Earlier Versions,” on page 49
- ◆ Chapter 4, “Adding Applications for Single Sign-On,” on page 55
- ◆ Chapter 5, “Managing SecureLogin,” on page 63
- ◆ Chapter 6, “Administering Scripts,” on page 93
- ◆ Chapter 8, “Setting Up Terminal Emulation,” on page 117
- ◆ Chapter 7, “Installing Terminal Servers,” on page 103
- ◆ Chapter 9, “Using Password Policies,” on page 145
- ◆ Chapter 10, “Troubleshooting SecureLogin,” on page 151

Additional Documentation

For documentation on Novell® SecretStore, see the Novell SecretStore [SecretStore Administration Guide](#).

For documentation on commands and example scripts for SecureLogin scripts, see [Script Commands](#).

For additional documentation on terminal emulators, see [Configuration Guide for Terminal Emulators](#).

Documentation Updates

For the most recent version of the *Novell SecureLogin Administration Guide*, see [SecureLogin](#) (<http://www.novell.com/documentation>) at the Novell Documentation Web page.

Documentation Conventions

In this documentation, a greater than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

1

Overview: Single Sign-On

This section provides information about the following:

- ♦ “SecureLogin As a Solution” on page 7
- ♦ “Understanding SecureLogin” on page 9

SecureLogin As a Solution

The Security Problem

Most organizations experience a major security problem—an abundance of applications and systems that require individual user authentication. This problem is worsening as the number of Internet applications increases. As more companies switch to eBusiness solutions, their authentication requirements change. Businesses now have new requirements for high-end authentication.

Networking dependencies have also brought about new security requirements for authentication, with an increasing need to identify the end user. This requirement is becoming increasingly difficult to meet.

Because technology continues to advance at a rapid pace, remembering usernames and passwords is also becoming increasingly difficult. An organization can no longer reasonably expect its users to remember long lists of username and password combinations, while still adhering to company security policies.

How Single Sign-On Fixes This Problem

Single sign-on fixes this problem by eliminating the need for users to remember their usernames and passwords beyond their initial network login. Single sign-on stores the usernames and passwords that each user needs and then automatically enters them for the user when required.

Single sign-on solves the security problem of users having to remember their login credentials. Single sign-on also increases productivity because users no longer need to enter usernames and passwords. The computer does it for them. As a result, single sign-on greatly reduces the number of calls to help desks concerning forgotten passwords.

The SecureLogin Solution

Novell® SecureLogin is comprised of multiple integrated security systems that provide authentication and single sign-on to networks and applications throughout an organization. The goal is to provide a single entry point to the corporate network and its resources for users, increase security, and improve compliance with corporate security policies.

The separate single sign-on modules (components) of SecureLogin are designed for generic Windows*, Internet, and terminal emulator applications. SecureLogin's unique modular design allows it to be compatible with most new applications.

Security is an important feature of SecureLogin. SecureLogin stores all user credentials encrypted in Novell eDirectory™ and optionally caches details in an encrypted format on the local workstation. The only user who can unlock the encrypted data is the user that the details are stored for. For example, a network administrator with all rights is not able to see a user's password for Internet banking.

SecureLogin is easy to use. Wizards, corporate scripts, predefined applications, and eDirectory enable you to intuitively configure—from a central point—SecureLogin for use in the corporate network. SecureLogin also includes a workstation administration tool that allows users to view their single sign-on details and, if you permit them to, add new applications and Web sites for single sign-on.

SecureLogin employs two methods of fault tolerance. One method uses local encrypted caching to ensure that network downtime does not affect single sign-on performance. Even if the corporate network is down, caching enables application logins to continue uninterrupted. A second method allows for scripting to cater to different login conditions and errors during login.

Local encrypted caching also enables SecureLogin to maintain single sign-on integrity for all mobile and remote users, regardless of network connectivity. If you permit them to, mobile users can update their single sign-on credentials when disconnected from the network and then update eDirectory with these details when the users are next attached.

Because SecureLogin is an eDirectory-enabled product, users can roam wherever eDirectory is. They can

- ◆ Log in from anywhere and get the same capabilities as if they were at their own desks.
- ◆ Log in and log off quickly, because they only authenticate to eDirectory, not to Windows itself.
- ◆ Roam the enterprise, logging into several different machines during the day.
- ◆ Work on a notebook or laptop in a disconnected mode, because their login credentials are saved to a local, encrypted cache.
- ◆ Securely use a shared, kiosk-type workstation, where many people log in temporarily for quick work and then log out.

Single sign-on has the following goals:

- ◆ Automate the repetitive manual login processes so that logging in is seamless to the user.
- ◆ Lower overall maintenance costs.
- ◆ Provide a secure, easy-to-use, fault-tolerant environment for logging in to Windows, Web, and mainframe applications.

Requirements for an Effective Implementation

A successful single sign-on system must meet the following seven requirements. SecureLogin was designed with these requirements in mind.

- ◆ Meet the changing needs of large organizations.
 - ◆ New software must be easily installed and configured for single sign-on.
- ◆ Easily accommodate mobile users.

Remote and roaming users must be able to access their single sign-on credentials and update them if necessary.

- ◆ Provide ease of management, rapid deployment, and fault tolerance.

The single sign-on system must run efficiently, be easy for users to operate, and be easy for you to control and maintain.

- ◆ Employ industry standards and open architecture.

The single sign-on system must be compatible with most existing software.

- ◆ Be secure.

The password storage and playback mechanism must not allow for stealing secrets. The usernames and passwords must be encrypted and stored in a secure database.

- ◆ Be cost effective.

The single sign-on system must save money and reduce the cost of ownership.

- ◆ Be seamless to the user.

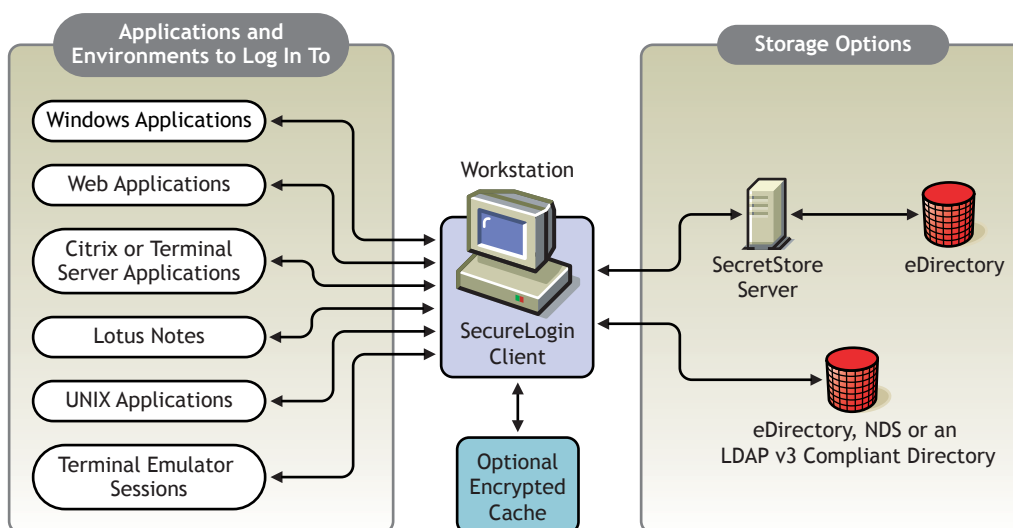
The second time a user logs in to an application, the application should look the same as the first time. Subsequent attempts to open the application should involve no user interaction for authentication.

Understanding SecureLogin

Novell SecureLogin runs on any platform that is running Novell eDirectory or previous versions of NDS[®], including NetWare[®] 4.12 and later. SecureLogin functionality depends on whether you are running an NDS version earlier than 8.5 or NDS 8.5 or later.

SecureLogin Architecture

SecureLogin is a suite of applications for authentication and single sign-on. As the following figure illustrates, it includes components for both client and server:



SecureLogin works by keeping a record of user authentication credentials and instructions for how to use those credentials. SecureLogin stores these credentials in the Directory, either directly or

through Novell's patented SecretStore[®] technology. At run time, SecureLogin detects login opportunities, retrieves the appropriate authentication credentials, and then automatically supplies those credentials.

The SecureLogin script language is a key feature of SecureLogin single sign-on. This language enables the product to be compatible with almost all network environments and applications. The script language has the following advantages:

- ◆ Enables you to define single sign-on methods for almost any Windows, mainframe, Internet, intranet, Terminal Server, or UNIX* application.
- ◆ Allows more sophisticated single sign-on to supported applications, including the ability to seamlessly handle several versions of one application.

This feature is especially important when you upgrade your applications.

SecureLogin data (for example, user credentials and application scripts) is stored and protected in the Directory. When used with eDirectory, SecureLogin can use SecretStore technology to provide an additional level of security. On startup, SecureLogin performs the following tasks:

- ◆ Locate these objects in the Directory.
- ◆ Cache their encrypted contents in memory (and optionally on disk) for later use by the workstation's SecureLogin single sign-on agent.

SecureLogin allows you to define which applications are to be enabled for single sign-on. This option gives you the following:

- ◆ Full control of which applications can be used for single sign-on.
- ◆ The ability to update the entire Directory database with a new application login script by updating a single object.

The corporate scripts are stored in a Container object rather than individual User objects. For users, the result is a less complex system. For you as the administrator, the improved login mechanisms provide the following:

- ◆ A greater level of accountability with increased productivity and security.
- ◆ A reduced load at the help desk because of significantly fewer password resets.

How SecureLogin Works

At Digital Airlines, users start ICA sessions on Citrix servers with the NetWare client installed. Upon initiating a session a user must be authenticated to eDirectory (or to NDS). Authentication is achieved by the company's GINA components passing the user's credentials to the NetWare Client interface. After the interface receives the credentials, the normal user level eDirectory transaction occurs between the Citrix server and eDirectory.

Upon authentication in eDirectory, ProLauncher starts SecureLogin (Proto.exe launches which in turn starts Combroker.exe). A call is made to eDirectory to acquire/synchronize assigned scripts and stored credentials.

Control is then handed off to the specified ICA application and any SecureLogin requests are handled from local cache. As the session ages, periodic refreshes of the SecureLogin store are attempted. The timeframe is adjustable by the administrator. This is the same call to eDirectory to acquire/synchronize assigned scripts and stored credentials. (This is where the -601 occurs.)

When a user ends the ICA application, ProLauncher ends SecureLogin.

Why this could happen.

- ◆ SecureLogin Error

Our application could have an issue interfacing to eDirectory at refresh time.

- ◆ eDirectory is down.

- ◆ eDirectory is busy.

If the replica that answers the call for SecureLogin information is busy, the request could timeout and the object would appear to be not available.

- ◆ eDirectory replicas are inconsistent.

It is possible that the Citrix servers are using a different replica ring than a user might. This could explain the disproportionate number of instances with the Citrix client.

- ◆ Rights are inadequate.

The workstation's or user's eDirectory rights could be inadequate for the requested item.

Script Language

The SecureLogin scripting language enables SecureLogin to be compatible with almost all network environments and applications. SecureLogin uses a script language to provide a flexible single sign-on and monitoring environment. For example, the SecureLogin Windows Agent watches for application login boxes. When a login box is identified, the agent runs a script to enter the username, password, and background authentication information.

The script language is used in individual application scripts to retrieve and enter the correct login details. These scripts are stored and secured within eDirectory to ensure maximum security, support for single-point administration, and manageability.

The script language can be used to automate many login processes, such as multi-page logins and login panels requiring other information (such as surname and telephone number) that can be stored in eDirectory. The script language also contains the commands required to automate password changes on behalf of users and request user input when it is required.

The scripting language has the following advantages:

- ◆ Enables you to define single sign-on methods for almost any Windows, mainframe, Internet, intranet, Terminal Server, or UNIX application.
- ◆ Allows more sophisticated single sign-on to supported applications, including the ability to seamlessly handle several versions of one application.

This feature is especially important when you upgrade your applications.

- ◆ SecureLogin data (for example, user credentials and application scripts) is stored and protected in the Directory.
- ◆ When used with eDirectory, SecureLogin can use SecretStore technology to provide an additional level of security.
- ◆ On startup, SecureLogin performs the following tasks:
 - ◆ Locates objects in the Directory.
 - ◆ Caches their encrypted contents in memory (and optionally on disk) for later use by the workstation's SecureLogin single sign-on agent.

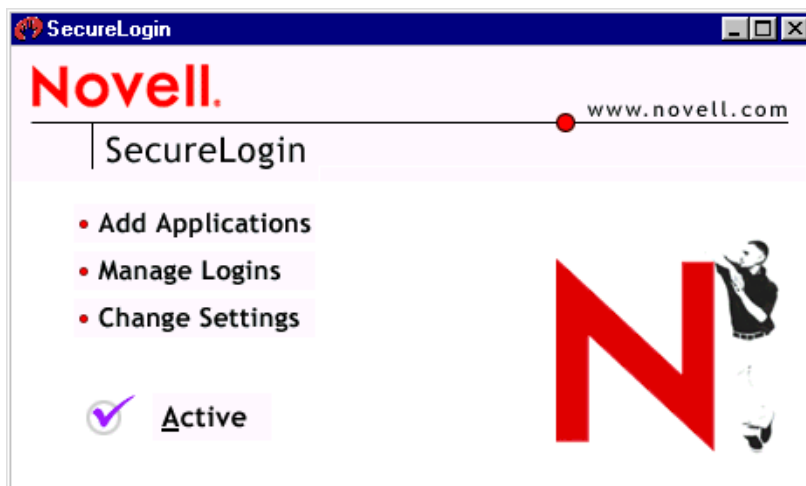
SecureLogin Components

SecureLogin provides the SecureLogin application that runs on users' workstations and also provides a snap-in to ConsoleOne®.

SecureLogin leverages your existing Net directory so that you can administer single sign-on solutions for applications, users, and the entire organization. With the SecureLogin administration tools, you can centrally manage users and corporate single sign-on applications and configurations.

The SecureLogin Application

Novell SecureLogin runs on the desktop. Users can use this tool to manage logins. The following figure illustrates the main window for this application:



This tool enables users to do the following at their workstations:

- ◆ Add new applications for single sign-on
- ◆ View existing applications and Web sites that single sign-on is enabled for.
- ◆ Modify passwords to existing single sign-on enabled sites.
- ◆ Control settings on how the product will behave

Using the ConsoleOne snap-in to eDirectory, you can enable or disable all of these functions for individual users and the entire organization.

You access this application through the Start > Programs > Novell SecureLogin option or by double-clicking the SecureLogin icon on the workstation's task bar. The following figure illustrates this icon:



SecretStore

When SecureLogin is used with eDirectory, you can use SecretStore, a patented Novell technology, to store your application passwords and other authentication credentials. SecretStore is a repository located within your eDirectory User object.

SecretStore provides an added level of protection and security to SecureLogin. Only the SecretStore server can access secrets, and each secret is stored separately, so that access to data is very compartmentalized and controlled. SecretStore also provides additional capabilities to deter would-be intruders, whether internal or external to your organization.

SecretStore runs on all eDirectory platforms: NetWare 5, NetWare 6, Windows NT/2000, Linux, and Solaris.

Terminal Launcher

Terminal Launcher enables you to log into any type of host that requires the user to log in using an emulator (for example, an ACF2 or RACF mainframe, a UNIX host, or a Cisco* router). Either you or the user configures Terminal Launcher to connect to the mainframe or host, wait for the login sequence, and then enter usernames and passwords.

Terminal Launcher enables you to easily launch terminal emulation sessions and to run a script within those sessions.

The script is stored within eDirectory, which makes it more secure than generic scripts that are written in a particular language for a particular emulator. These scripts are designed to be compatible with many different emulators.

With the use of corporate scripts, Terminal Launcher is very powerful. It can be used to provide shortcut icons to mainframe or UNIX applications, removing the need for user intervention.

The following figure illustrates Terminal Launcher:



You access Terminal Launcher from Start > Programs > Novell SecureLogin.

Corporate Login Scripts

SecureLogin is designed for large networks. It supports the ability to use eDirectory to centralize the setup of the single sign-on applications. This feature is referred to as Corporate Login Scripts.

A corporate login script can be stored in either a file system or in a Container object located in eDirectory. This feature gives you the ability to write and define single sign-on scripts once for the whole organization, while still allowing for customized subordinate containers and User objects. This customization significantly reduces the effort and complexity of enterprise deployment.

If a subordinate object has a different script for the same application defined locally, the local copy will be used instead of the version that is on the higher object. If a script is defined on a User object with the same name as a script defined on a Container object, or if there are two scripts with the same name on different level Container objects, the script from the subordinate object will always be used instead of the script in the higher level object. This strategy allows for specialization in corporate scripts.

For more detail on scripting, see [Chapter 6, “Administering Scripts,” on page 93](#).

Window Finder

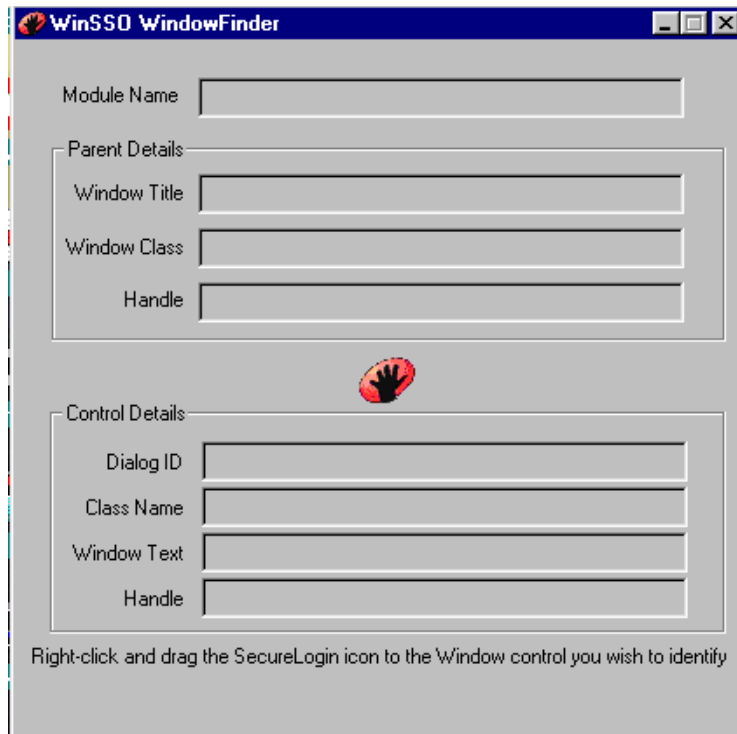
Window Finder is a component for Windows applications. While you are creating a script, Window Finder enables you to find out information about the window that the login box is on. Window Finder shows you the name of a control button (for example, OK or Next) and reveals the control ID number.

This component works with

- ◆ All Windows applications that have generic username and password prompts

- ◆ Most specialized applications that have very complex logins

The following figure illustrates Window Finder:



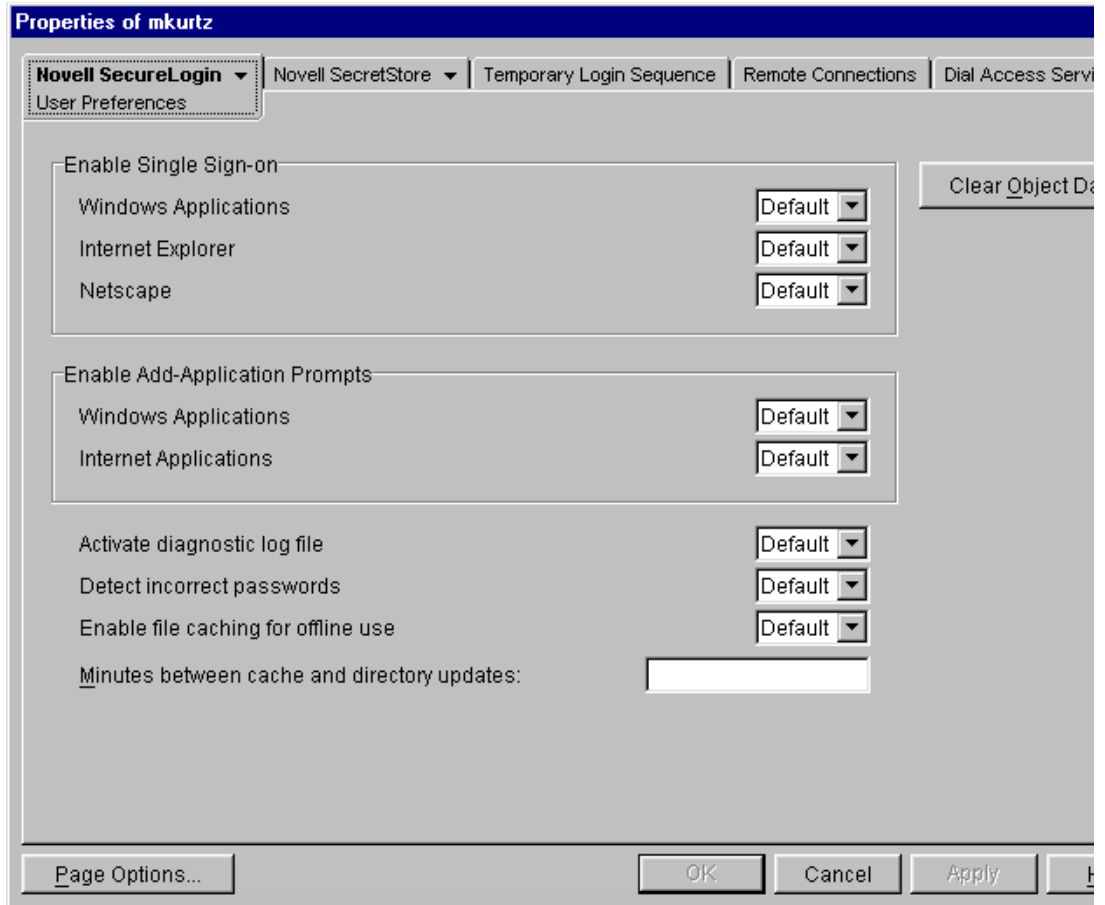
You access Window Finder by clicking Start > Programs > Novell SecureLogin > Window Finder.

The Snap-In to ConsoleOne

SecureLogin 3.0 provides a snap-in to ConsoleOne. Using ConsoleOne, you are able to do the following:

- ◆ Set SecureLogin preferences for the users at either the User object level or at the Container object level
- ◆ Manage logins
- ◆ Set administrative preferences

The following figure illustrates the ConsoleOne window for setting user preferences:



Internet Browsers

The Microsoft* Internet Explorer and Netscape* components enable applications that are accessed through these browsers to use single sign-on. Depending on a workstation's configuration, the browsers might behave differently.

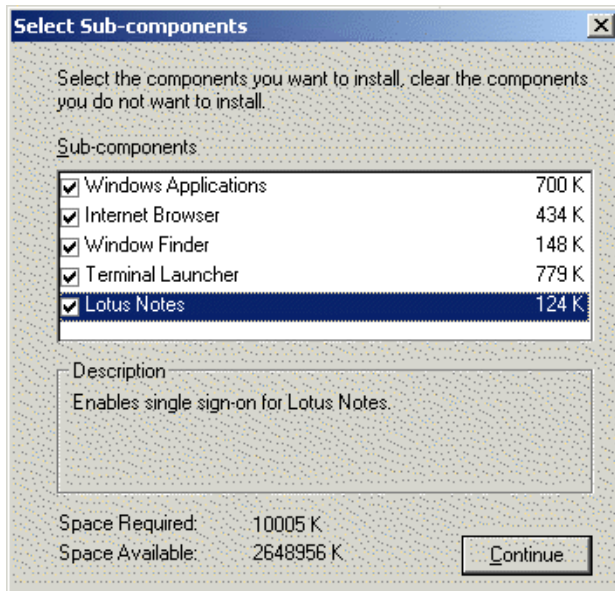
These components also enable sites using http dialogue authentication to use single sign-on.

Lotus Notes

The SecureLogin Lotus Notes* component enables you to use single sign-on with Lotus Notes. This component is a more specialized version of the Windows applications single sign-on component and is designed so that you do not notice you have switched over to single sign-on (apart from the lack of login windows).

When you install SecureLogin on a workstation, the installation program looks for the HKEY_LOCAL_MACHINE\SOFTWARE\Lotus\Notes key in the Windows registry. If Lotus Notes is installed, a Lotus Notes SecureLogin file (pronotes.dll) is automatically installed. This file tightly integrates with the Lotus Notes authentication system.

The following figure illustrates the Lotus Notes option that is available during installation:



This option is only displayed if the installation program detects that Lotus Notes is installed on the workstation.

The installation program also updates the Lotus Notes notes.ini file by adding the following:

```
EXTMGR_ADDINS=pronotes.dll
```

After installation, the next time you authenticate to Lotus Notes you actually type your password into a SecureLogin panel designed to look like the Lotus Notes password box. This will be the last time you need to enter your password into Notes.

The next time you are required to authenticate, SecureLogin communicates with Lotus Notes in the background. The password box to log in never appears. At the end of the password expiration period, SecureLogin can prompt for a new password or automatically populate the password field.

SecureLogin supports password expiration in Notes and, as with all applications, can be set up to automatically generate a random password, based on a password policy. In addition to controlling single sign-on, this component supports

- ◆ Multiple ID files for each user
- ◆ The ability to exclude certain administrative IDs from being enabled for single sign-on

Mobile Single Sign-On

Taking advantage of eDirectory architecture, SecureLogin allows users to roam with their authentication details. Because there are no workstation dependencies, users can move freely from office to office. Their credentials follow them.

By using the local encrypted cache, SecureLogin also allows notebook users access to single sign-on.

2

Installing SecureLogin

This section provides information about the following:

- ♦ “Before Installing SecureLogin on Workstations” on page 19
- ♦ “Installing SecureLogin on an Active Directory Server” on page 27
- ♦ “Setting Up SecureLogin for NT 4 Domains” on page 39
- ♦ “Installing SecureLogin” on page 43
- ♦ “Granting Rights” on page 46
- ♦ “Installing Administration Tools” on page 49

An administrator of Novell® eDirectory™ or earlier versions of NDS® should do the first installation so that the schema is extended properly.

To install Novell SecretStore®, see the *Novell SecretStore Administration Guide*.

Before Installing SecureLogin on Workstations

Before installing SecureLogin on workstations, review supported platforms, determine a type of installation, and complete pre-installation tasks.

Supported Platforms

Novell SecureLogin 3.0 supports the following platforms. Latest support packs are recommended for all platforms.

- ♦ Servers
 - ♦ Novell Directory Services (NDS) on NetWare® 4.2 or NetWare 5.x or later
 - ♦ Novell eDirectory on NetWare, Windows* NT*, or Windows 2000

In the non-SecretStore mode, SecureLogin runs against eDirectory on any platform.

SecureLogin 3.0 only supports ConsoleOne® 1.3.2 or later.

- ♦ Browsers: Internet Explorer 5.01 or later or Netscape* 4.7.x
- ♦ Workstations: Windows 95 OSR2B, Windows 98 SE, Windows 2000, Windows XP, or Windows NT 4.0 or later

Windows 95 and Windows 98 workstations should have Novell Client™ 3.21 or later.

Windows 2000 workstations must have Novell Client 4.71 or later.

The SecureLogin snap-in to ConsoleOne requires ConsoleOne 1.3.2 or later. The installation program can install version 1.3.3.

Types of SecureLogin Installations

When you install SecureLogin on a workstation, you must select one of the following installations:

- ◆ Novell eDirectory with SecretStore
- ◆ Novell eDirectory
- ◆ LDAP v3.0 Compatible Directory
- ◆ Other

Before installing on a workstation, complete necessary pre-installation tasks.

Novell eDirectory with SecretStore

The Novell eDirectory with SecretStore option installs SecureLogin onto networks that are running one of the following:

- ◆ eDirectory on NetWare 5.1, NetWare 6, or Windows NT/2000 servers
- ◆ NDS (NetWare 5.1 or later)

This option uses Novell's patented SecretStore client/server system to provide the highest possible level of security for user login data. SecretStore requires server components on the eDirectory server and SecureLogin client software on workstations.

Before Installing SecureLogin with SecretStore

If you plan to use SecureLogin with SecretStore, complete the following tasks before installing SecureLogin on a workstation:

- 1** Install SecretStore on a server.
See [Installing SecretStore](#) in the *Novell SecretStore Administration Guide*.
- 2** Extend the NDS or eDirectory schema.
See [“Extending the eDirectory Schema” on page 26](#).
- 3** Prepare the workstation.
See [“Preparing the Workstation” on page 26](#).
- 4** Migrate earlier versions of Novell Single Sign-On or Novell SecureLogin.
- 5** Ensure that the workstation's current primary tree and server connections are set to the tree in which the SecretStore service has been installed.

Novell eDirectory

The Novell eDirectory option installs SecureLogin onto networks that are running NDS[®] (NetWare 4.2 or later) or eDirectory.

This option provides secure, centralized storage of user login data by performing encryption once on the workstation before the data is saved to eDirectory. No server components are installed for this option. The first installation of client software must be done by an eDirectory administrator to extend the schema and assign user rights.

Before Installing SecureLogin with eDirectory

If you plan to use SecureLogin with eDirectory, complete the following tasks before installing SecureLogin on a workstation:

- 1** Extend the NDS or eDirectory schema.
See [“Extending the eDirectory Schema” on page 26.](#)
- 2** Prepare the workstation.
See [“Preparing the Workstation” on page 26.](#)
- 3** Migrate earlier versions of Novell Single Sign-On or Novell SecureLogin.
- 4** Ensure that the workstation's current primary tree and server connections are set to the tree in which the SecretStore service has been installed.

LDAP v3.0 Compatible Directory

The LDAP option installs SecureLogin into LDAP v3.0 directory environments (for example, Novell eDirectory 8.5 or later).

This option does not require the Novell Client for Windows.

Before Installing SecureLogin with LDAP

If you plan to use SecureLogin with LDAP, complete the following tasks before installing SecureLogin on a workstation:

- 1** Extend the LDAP directory schema.
See [“Extending the eDirectory Schema” on page 26.](#)
- 2** Grant rights.
See [“Granting Rights” on page 46.](#)

- 3** Set up LDAP mappings.

Before LDAP client support can be used, you must map NDS or eDirectory attribute names to LDAP names.

The LDAP v3.0 client option supports servers that have the following:

- ♦ Novell eDirectory 85.0.1 or later.
- ♦ LDAP support installed and running.

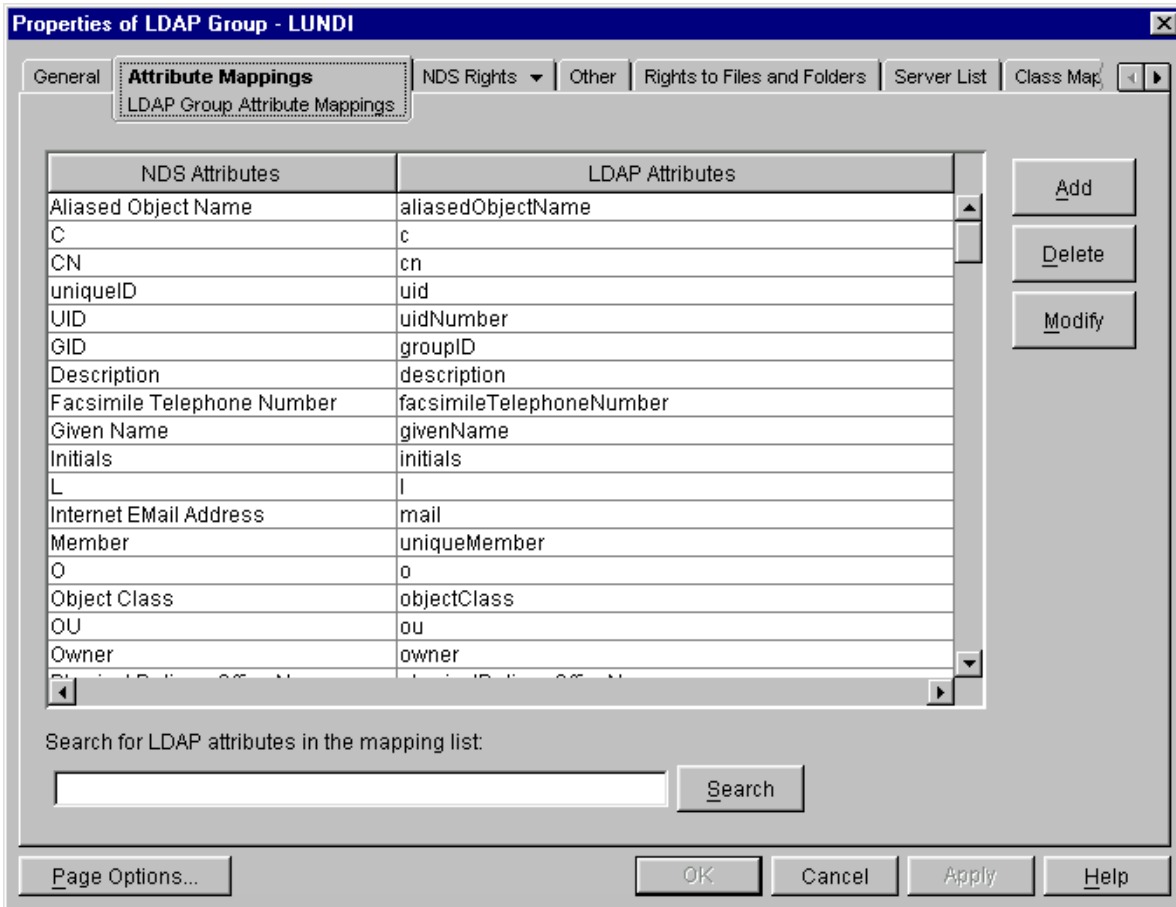
3a Establish a Novell Client connection to the NDS or eDirectory server where you want to run LDAP compatibility mode.

3b From that client connection, launch ConsoleOne.

3c Select the LDAP Group object for your server.



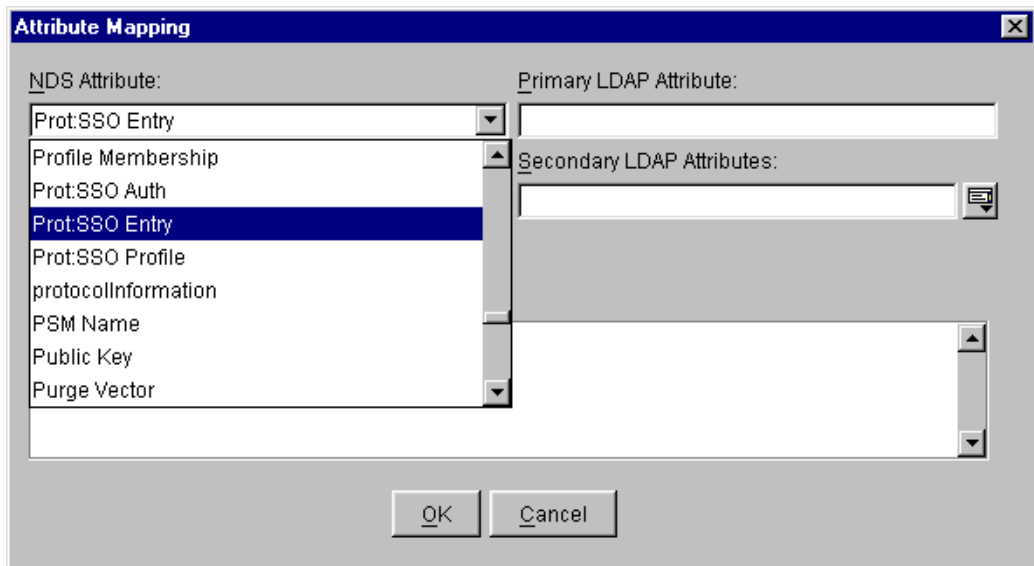
3d Display the Attribute Mappings tab by clicking Properties > Attribute Mappings.



If you can't locate this tab, you must install the LDAP snap-in to ConsoleOne. Download the snap-in from <http://download.novell.com>. Select ConsoleOne Snap-ins > On NetWare > NDS eDirectory 8.5 Snap-in.

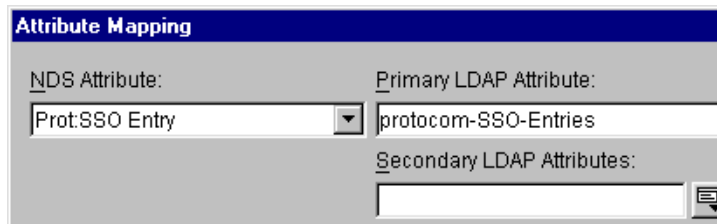
3e Click Add.

3f From the NDS Attribute drop-down list, select the Prot:SSO Entry attribute.



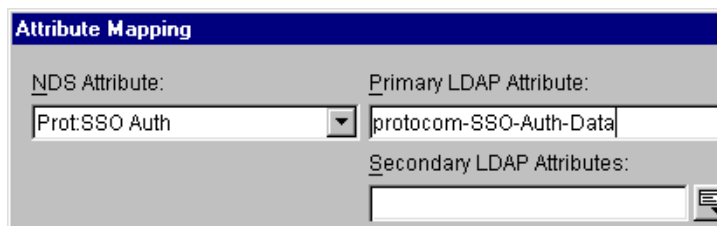
If the Prot:SSO Entry attribute is unavailable, run NDSSchema.exe or LDAPSchema.exe. These files are in the securelogin/tools directory.

- 3g** Map the Prot:SSO Entry attribute to protocom-SSO-Entries, as indicated in the following figure.



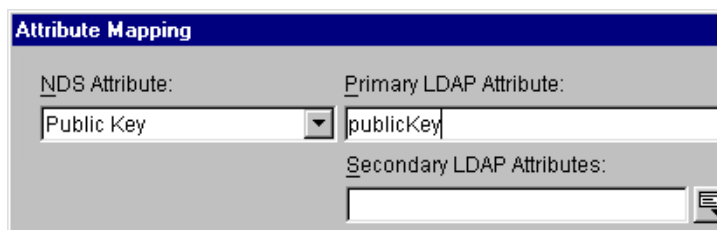
The screenshot shows the 'Attribute Mapping' dialog box. It has a title bar 'Attribute Mapping' in a blue header. Below the title bar, there are three fields: 'NDS Attribute:' with a dropdown menu showing 'Prot:SSO Entry', 'Primary LDAP Attribute:' with a text box containing 'protocom-SSO-Entries', and 'Secondary LDAP Attributes:' with an empty text box and a small icon to its right.

- 3h** Similarly, map the Prot:SSO Auth attribute to protocom-SSO-Auth-Data.



The screenshot shows the 'Attribute Mapping' dialog box. It has a title bar 'Attribute Mapping' in a blue header. Below the title bar, there are three fields: 'NDS Attribute:' with a dropdown menu showing 'Prot:SSO Auth', 'Primary LDAP Attribute:' with a text box containing 'protocom-SSO-Auth-Data', and 'Secondary LDAP Attributes:' with an empty text box and a small icon to its right.

- 3i** Similarly, map the Public Key attribute to publicKey.



The screenshot shows the 'Attribute Mapping' dialog box. It has a title bar 'Attribute Mapping' in a blue header. Below the title bar, there are three fields: 'NDS Attribute:' with a dropdown menu showing 'Public Key', 'Primary LDAP Attribute:' with a text box containing 'publicKey', and 'Secondary LDAP Attributes:' with an empty text box and a small icon to its right.

- 3j** Click Apply and then click Close.

- 3k** Refresh the LDAP server.

If you are using ConsoleOne, right-click the LDAP Server object, click Properties, and then click Refresh NLDAP Server Now.

If you are using Novell iManager, click LDAP Management, click LDAP Overview, click View LDAP Servers, select the LDAP server, and then click Refresh.

- 4** Install a Trusted Root certificate by copying RootCert.der from sys:\public to c:\Program Files\novell\securelogin.

After you select the LDAP option during a SecureLogin installation, the installation program copies the RootCert.der file to the c:\program files\novell\securelogin directory. This is a placeholder file. You must replace RootCert.der with an actual Trusted Root certificate file. Users must browse to this file and use it for an SSL LDP bind on their client workstations.

When eDirectory is installed on NetWare, RootCert.der is exported to the sys:\public directory. When eDirectory is installed on Windows NT or Windows 2000, RootCert.der is exported to the winnt\profiles\administrator\recent directory.

If `sys:\public` or `winnt\profiles\administrator\recent` is unavailable, you can create, export, and copy a `TrustedRootCert.der` file:

1. Create and export `TrustedRootCert.der` from the certificate by using the PKI snap-ins in `ConsoleOne`. (`TrustedRootCert.der` is the default name for the exported copy of the `RootCert.der` file.)
2. Copy `TrustedRootCert.der` from the location you put it into the `\securelogin\client\Program Files\novell\securelogin` directory (in a copy of the CD image).
3. Run `setup.exe` from the `\securelogin\client` directory.

5 Provide information for users.

When using the LDAP connectivity option, the user must provide LDAP server information during the first login. For subsequent logins, this information is automatically saved and entered into the login dialog box.

You must provide users with the following:

- ◆ The registered DNS name or IP Address
- ◆ The IP Port for secure LDAP

By default, this is port 636. When entered, it is saved in the workstation's registry for subsequent logins.

To simplify the initial LDAP login, you can preconfigure these registry values as part of an automated client deployment. To do this, modify the `HKEY_CURRENT_USER` registry key.

The following lines contain a sample registry entry:

```
HKEY_CURRENT_USER\Software\Protocom\SecureLogin\LDAP Settings
"PrimaryHost"="151.155.164.77"
"SecondaryHost"="151.155.165.88"
"PrimaryPort"=dword:0000027c
"SecondaryPort"=dword:0000027c
"SSL Cert File"="C:\\TrustedRootCert.der"
"Context1"="O=Novell"
```

Other

The Other option enables you to install a standalone version, Microsoft Active Directory* Server Interface (ADSI), or Microsoft NT 2000 Domains.



Standalone

The Standalone option installs SecureLogin on a workstation and runs without eDirectory synchronization. This option uses only local cache files.

Select this option to demonstrate or evaluate the product.

Active Directory Server

Before installing SecureLogin on a workstation for Active Directory, set up SecureLogin for an Active Directory server. See [“Installing SecureLogin on an Active Directory Server” on page 27](#).

A Microsoft NT or Windows 2000 Domain

If you have a mixed Windows NT/2000 environment, follow instructions in [“Setting Up SecureLogin for NT 4 Domains” on page 39](#). If all users are connecting to a Windows 2000 server running ADS, follow instructions in [“Installing SecureLogin on an Active Directory Server” on page 27](#).

Then install SecureLogin on workstations.

Extending the eDirectory Schema

For SecureLogin to be able to save user single sign-on information, the eDirectory schema must be extended. Therefore, for the first installation of SecureLogin into an eDirectory tree, run NDSSchema.exe. (You only extend the eDirectory tree schema once for SecureLogin.)

NDSSchema.exe also grants existing users rights to the SecureLogin attributes on the User object. This file is in the securelogin\tools directory. You can run NDSSchema.exe multiple times to grant rights to users that you create after installing SecureLogin.

To extend the schema of a given tree, you must have sufficient rights over the [root] of the tree.

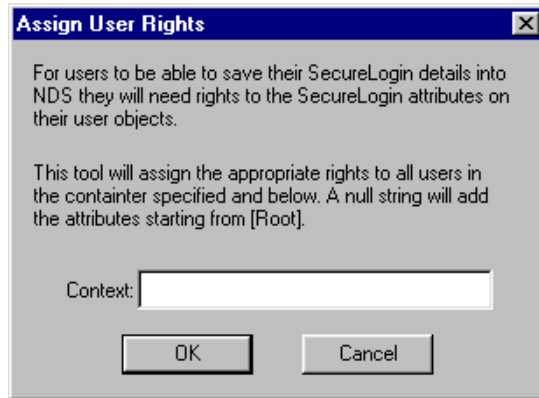
WARNING: Don't run NDSSchema.exe from a Windows 98 workstation. SecureLogin doesn't support using Windows 98 to run NDSSchema.exe.

- 1 At the securelogin\tools directory, run NDSSchema.exe.

The extension may take some time to filter throughout your network, depending on the size of your network and the speed of the links.

- 2 Enter an eDirectory context so that SecureLogin can assign rights to User objects.

You will be prompted to define a context where you want the User objects' rights to be updated, allowing users access to their own single sign-on credentials. The following figure illustrates this prompt:



If the installation program displays a message similar to -601 No Such Attribute, you have probably entered an incorrect context or included a leading dot in the context.

Preparing the Workstation

For Windows 95/98, Windows NT/2000, and Windows XP Pro workstations, install required software as listed in the following table:

Type of Installation	Required Software
eDirectory with SecretStore	<ul style="list-style-type: none">◆ Novell International Cryptographic Infrastructure (NICI) Client 2.02 or later◆ The latest Novell Client
eDirectory	The latest Novell Client
LDAP	None
Standalone	None
Other	SP6A for NT 4

To enable support for NMAS NDS Password disconnected login, set the registry key “NMAS required for disconnected mode” (a DWORD value) to 1. This key resides under HKEY_CURRENT_USER\Software\Protocom\SecureLogin.

To disable support, delete the DWORD value or set it to 0.

Installing SecureLogin on an Active Directory Server

The following information assumes that you have installed the Microsoft* Windows 2000 Server family operating systems (including Active Directory) on at least one domain controller in your network.

Management of the schema is restricted to a group of administrators called schema administrators. The Active Directory Schema snap-in allows schema administrators to manage the Active Directory schema by doing the following:

- ♦ Creating and modifying classes and attributes.
- ♦ Specifying which attributes are indexed and which attributes are to be catalogued in the global catalog.

As a schema administrator, you won't perform schema management tasks frequently. Observe three safety precautions that control and limit schema modification:

- ♦ By default, all domain controllers permit Read access to the schema. A registry entry must be set on a domain controller to permit Write access to the schema on that domain controller.
- ♦ The schema object is protected by the Windows 2000 Security model. Therefore, administrators must be given explicit permissions or be members of the Schema Administrators group to make changes to the schema.
- ♦ Although Active Directory is based on a multi-master administration model, some operations support only a single master. One of these operations is schema management. Only one domain controller can write to the schema at any given time. This role is known as Schema Floating Single Master Operations (FSMO).

To manage the schema, you must be connected to the schema FSMO. By default, the schema snap-in is targeted to the schema FSMO role.

Adding Administrative Tools for Active Directory

The following procedures assume that you are logged in as an administrator with the required permissions to manage the schema.

- 1** Click Start > Settings > Control Panel > Add/Remove Programs.
- 2** Click Windows 2000 Administration Tools > Change, then click Next.
- 3** Click Install All Administrative Tools, then click Next.
- 4** After components and files are installed, click Finish, then click Close.

Starting the Active Directory Schema Snap-In

The Active Directory Schema snap-in is a Microsoft Management Console (MMC) tool. Because schema management is not frequently performed, there is no saved Schema console or Administrative Tool on the Administrative Tools menu. You must manually load the Schema Manager into MMC.

Run the following procedure on the domain controller that contains the schema.

- 1** Click Start > Run.
- 2** In the Open box, type **MMC** and then click OK.
- 3** From the Console drop-down list, click Add/Remove Snap-In > Add.

- 4** Click Active Directory Schema > Add.
- 5** Click Active Directory Users and Computers > Add.
- 6** Click Close, then click OK.
- 7** Save the MMC containing the schema snap-in.
 - 7a** From the Console drop-down list, click Save As.
 - 7b** Type a name for the saved console (for example, Schema.msc).
 - 7c** Click Save.

Extending the Active Directory Management Schema

You can transfer the schema FSMO from one server to another. However, if you have installed a single Windows 2000 domain controller in your network, this procedure is unnecessary. By default, that single domain controller handles the schema FSMO role.

Transferring the Schema FSMO to Another Domain Controller

To transfer the schema FSMO to another domain controller:

- 1** From the left pane of the MMC console, right-click Active Directory Schema.

The following graphic illustrates Active Directory Schema in the directory structure:
- 2** Click Change Domain Controller.
- 3** (Conditional) If the name in the Current DC field is not the target server, click Specify Name, type the name of the target domain controller, then click OK.

The following figure illustrates the Current DC field:



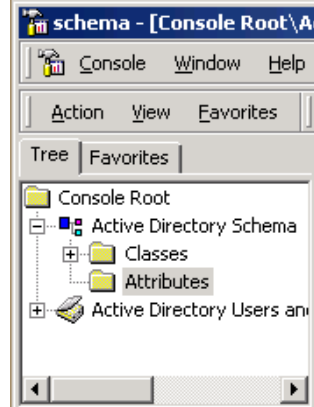
- 4** From the left pane, right-click Active Directory Schema, click Operations Master, then click Change.
- 5** Click OK to confirm that you want to change the Operations Master.
- 6** When you receive the message that the Operations Master was successfully transferred, click OK.

Verifying the Domain Controller

To verify that you have selected the correct domain controller:

- 1 From the left pane of the MMC console, right-click Active Directory Schema and then click Change Domain Controller.

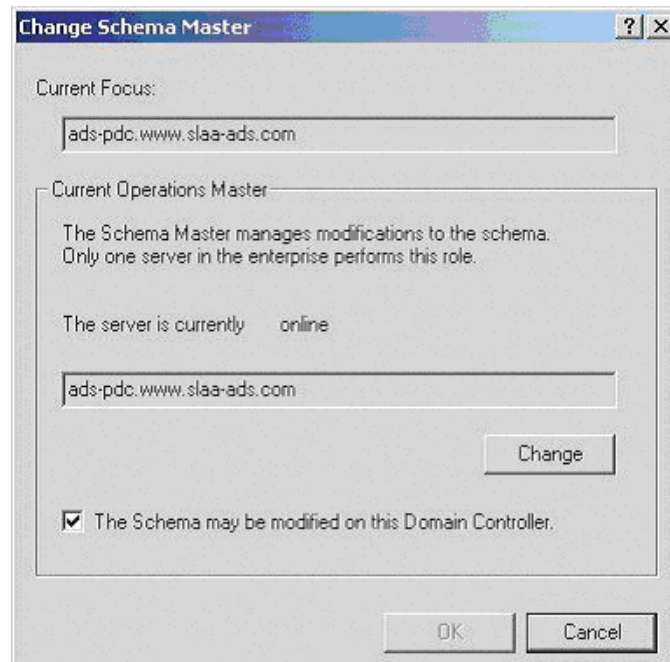
The following figure illustrates Active Directory Schema in the directory structure:



- 2 Verify that the Current DC field lists the domain controller that you are currently working on and then click OK.
- 3 From the left panel, right-click Active Directory Schema and then select Operations Master.
- 4 Check the Schema May Be Modified on This Domain Controller check box and then click OK.

This check box sets a registry entry that permits schema updates. The server automatically detects the change to this registry. You don't have to restart the server to permit the schema to be updated.

The following figure illustrates this check box:



Extending the Active Directory Schema

To store information such as a user's credentials, application scripts, preferences and corporate configuration, you must extend the Active Directory schema to accommodate three new object attributes.

1 Run adsschema.exe.

This file is on the Novell SecureLogin CD in the \securelogin\tools directory.

When you run adsschema.exe on the server that is the FSMO master, adsschema.exe adds three attributes to the schema:

- ◆ ProtAuthMethods

This attribute is only for a User object. It is an octet-string type.

- ◆ 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 userIDs and passwords
- ◆ Specific preferences and application definitions at the User object
- ◆ Corporate application definitions and preferences at the Container and Organizational Unit objects

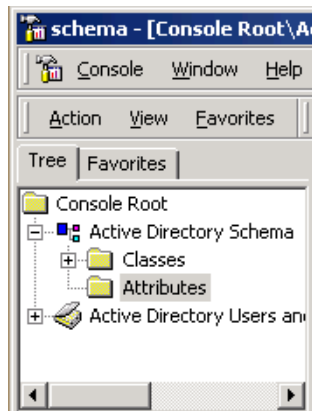
2 Verify that the schema has been extended.

2a Close and restart MMC.

After extending the schema, you must close and restart MMC before you can verify that the schema has been extended

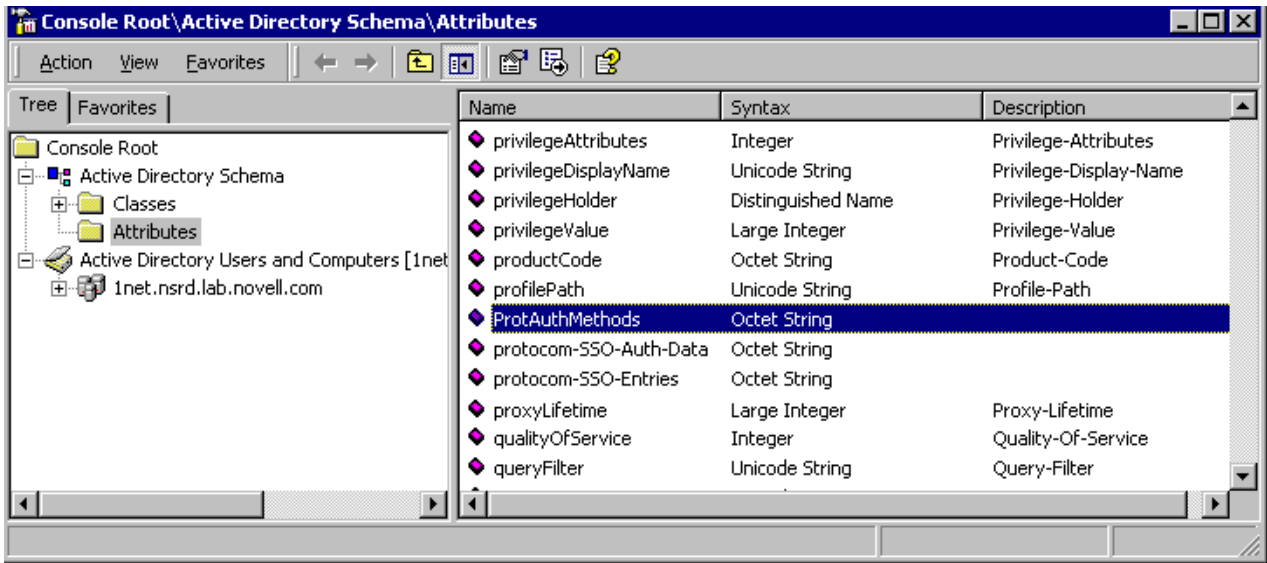
2b In the MMC tool, navigate to the Attributes folder.

The following figure illustrates the Attributes folder:



2c Identify the three attributes.

Ensure that ProtAuthMethods, protocom-SSO-Auth-Data, and protocom-SSO-Entries appear in the ADS list of attributes. The following figure illustrates these attributes that have extended the schema:

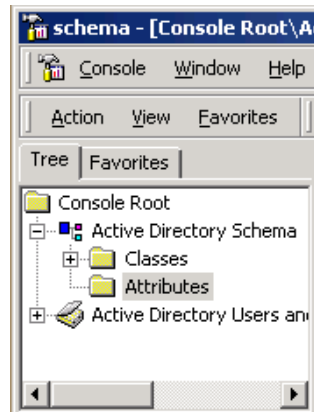


Replicating Three Attributes

To enable other servers to have the ProtAuthMethods, protocom-SSO-Auth-Data, and protocom-SSO-Entries attributes, you must replicate the attributes.

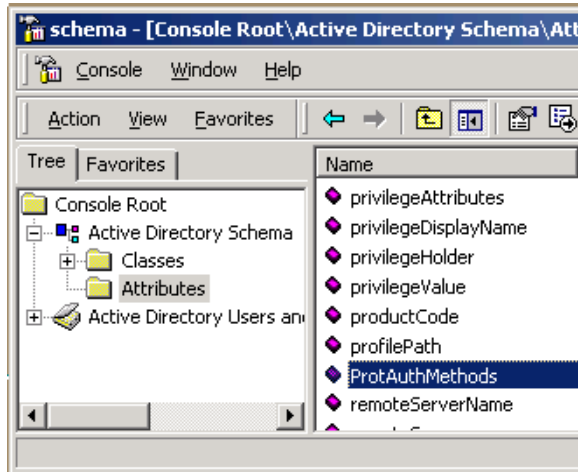
- 1 In the MMC tool, navigate to the Attributes folder.

The following figure illustrates the Attributes folder:

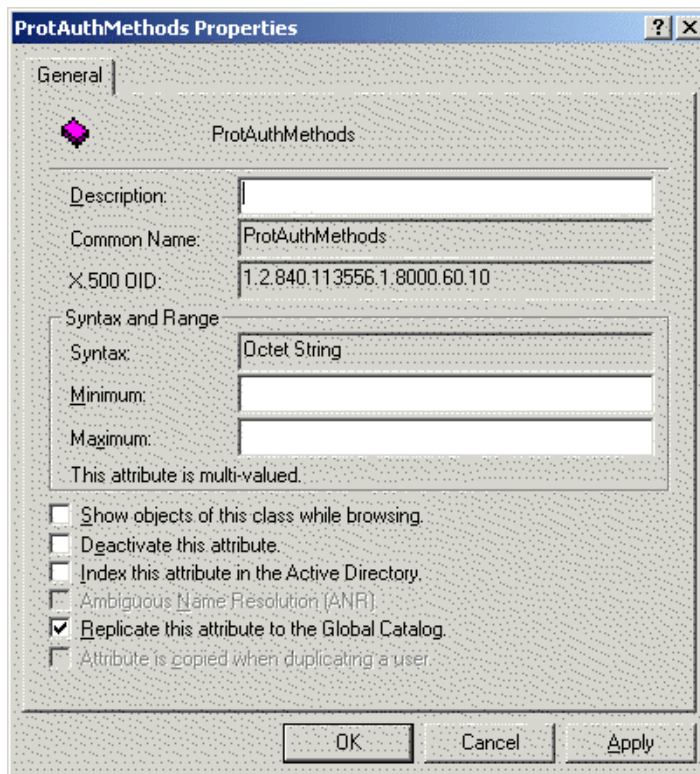


- 2 Right-click the ProtAuthMethods attribute, then click Properties.

The following figure illustrates the ProtAuthMethods attribute:



- 3** Check the Replicate This Attribute to the Global Catalog check box, then click OK.
The following figure illustrates this check box:



- 4** Repeat this process for the protocom-SSO-Auth-Data and protocom-SSO-Entries attributes.
- 5** Shut down and restart the management console.

Active Directory does not incorporate the new attributes until the management console is restarted.

Assigning Rights to User Objects

To use SecureLogin, users must have Read and Write rights to the ProtAuthMethods, protocom-SSO-Auth-Data, and protocom-SSO-Entries attributes on their User object. These rights enable users to add configuration data (for example, a passphrase) and create logins.

Default rights are set when SecureLogin is installed and the schema is extended for the first time.

If you don't assign rights to SELF, users are unable to read or write SecureLogin attributes.

To assign rights:

- 1** Bring up the MMC snap-in.
 - 1a** Click Console, then click Open.
 - 1b** Select the profile name that you saved in [Step 7 on page 28](#).
 - 1c** Click Open.
- 2** Click Active Directory Users and Computers and then click the domain name (for example, inet.nsrld.lab.vmp.com) > Users.
- 3** Right-click a container, click Delegate Control, then click Next.

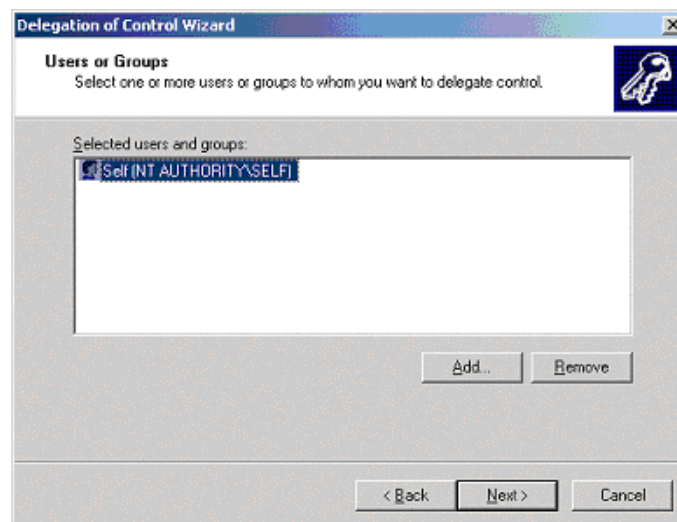
For example, you can select the Users container. Active Directory automatically creates this predefined container. On the other hand, you can select a container that you have created (for example, RDlab).

This step is necessary for every container that you want rights to apply to.

If Container objects (for example, OU objects) contain users in subcontainers, you must set up the same rights as the ones assigned to Active Directory's built-in Users container.

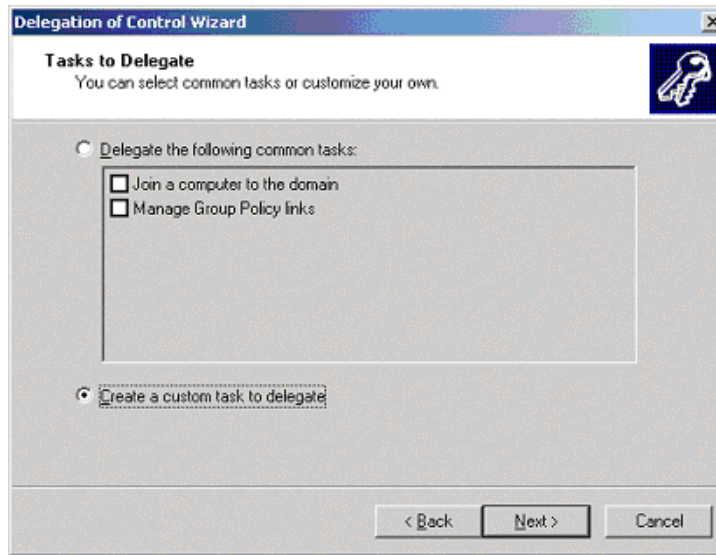
If branches exist in your Active Directory tree, ensure proper rights by assigning rights for each branch or by assigning rights globally at the Root.

The following figure illustrates the Selected Users and Groups list box:



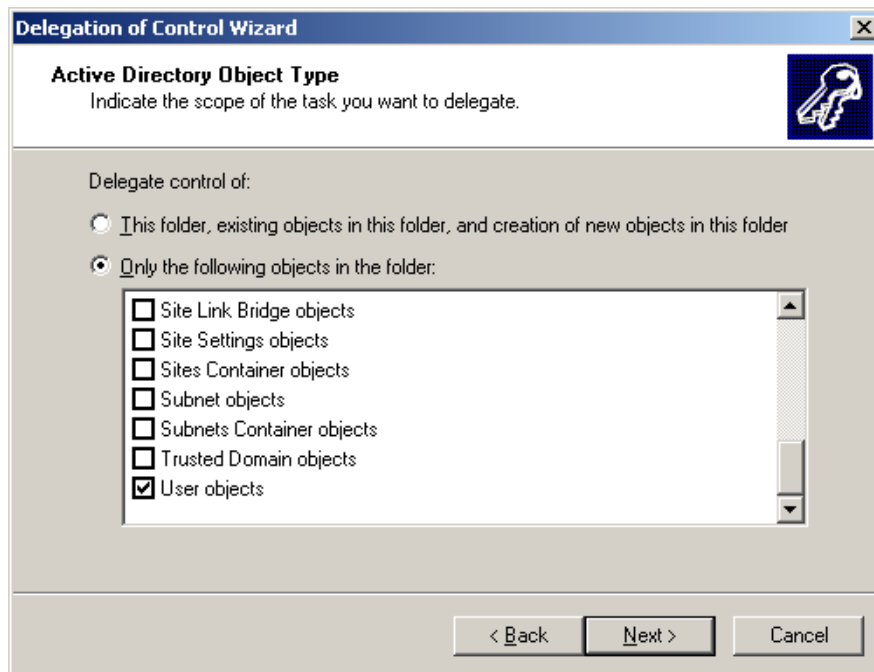
- 4** Click Add, then select SELF.
- 5** Click Add, then click OK.
- 6** (Conditional) Click Create a Custom Task to Delegate, then click Next.

If you selected the predefined Users container, skip this step. The following screen won't appear. However, if you selected a container that you created (for example, RDlab), the following screen appears.



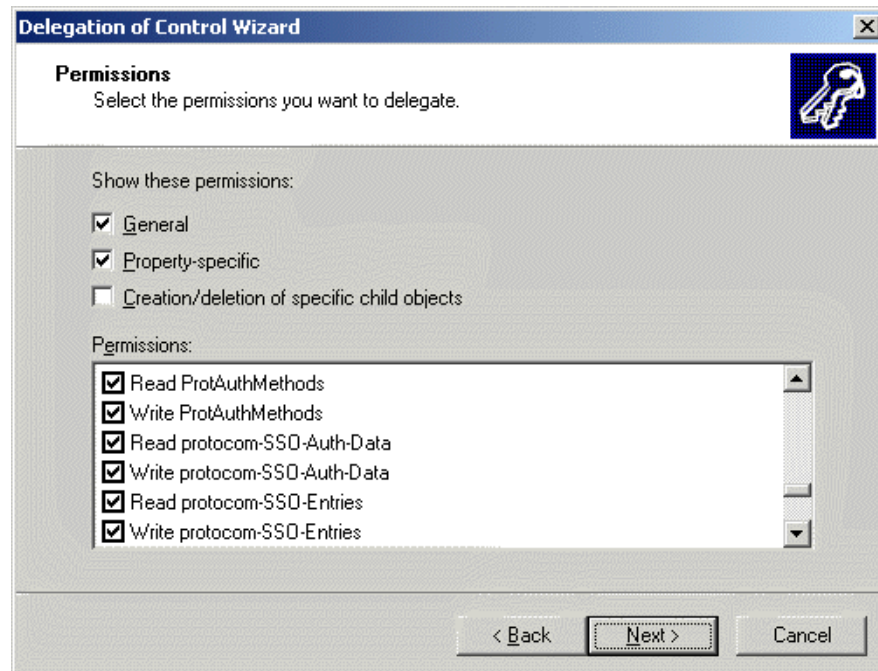
- 7** In the Active Directory Object Type window, click Only the Following Objects in the Folder, check the User Objects check box, then click Next.

The following figure illustrates this option:



- 8** Set permissions on new schema attributes.
 - 8a** Under Show These Permissions, check the General and Property-Specific check boxes.
 - 8b** In the Permissions list box, check the Read and Write check boxes for the ProtAuthMethods, protocom-SSO-Auth-Data, and protocom-SSO-Entries attributes.

The following figure illustrates these check boxes:



- 9 Click Next, then click Finish.

Assigning User Rights to an Organizational Unit

In addition to setting rights for User objects, you must set rights so that users can read corporate objects (for example, corporate scripts and serverPolicyOverride objects). Users can then inherit and use objects that you set up specifically for target users.

To accomplish this, you set Read and Write permissions for the prot-SSO-Entries attribute.

Settings and corporate scripts.

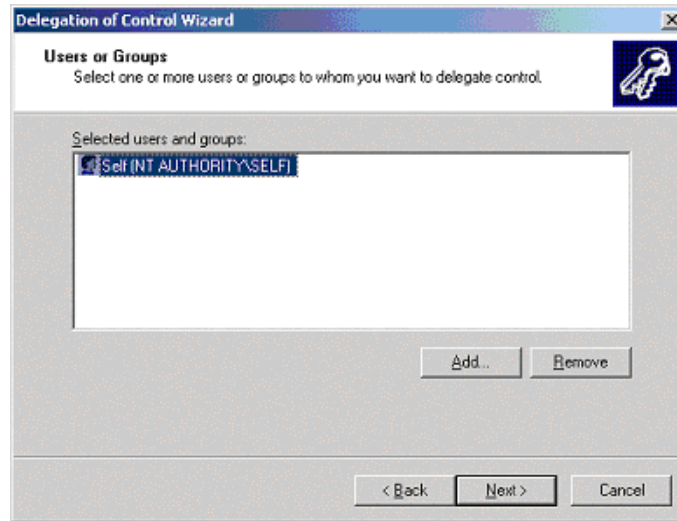
- 1 Bring up the MMC snap-in.
 - 1a Select Console, then click Open.
 - 1b Select the profile name that you saved in [Step 7 on page 28](#).
 - 1c Click Open.
- 2 Select Active Directory Users and Computers, select the domain name (for example, inet.nsrld.lab.vmp.com), and then click Users.
- 3 Right-click the container that you want to apply rights to, select Delegate Control, then click Next.

For example, you can select the Users container. Active Directory automatically creates this predefined container. On the other hand, you can select a container that you have created (for example, RDlab).

If Container objects (for example, OU objects) contain users in subcontainers, you must set up the same rights as the ones assigned to Active Directory's built-in Users container.

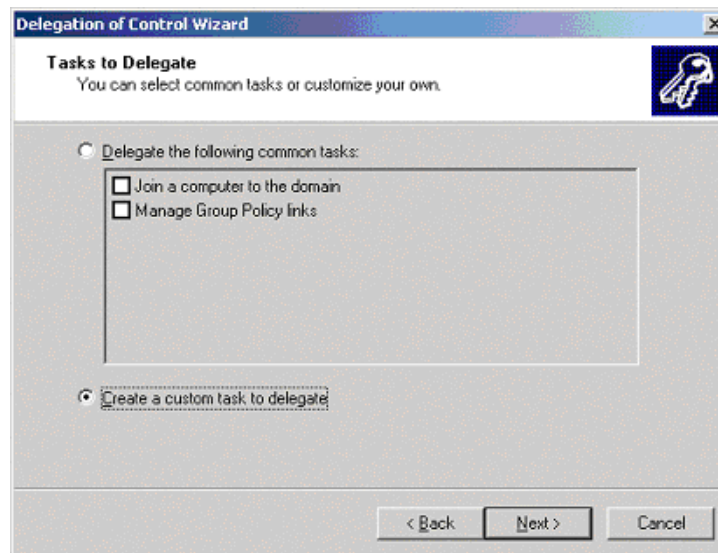
If branches exist in your Active Directory tree, ensure proper rights by assigning rights for each branch or by assigning rights globally at the Root.

The following figure illustrates the Selected Users and Groups list box:

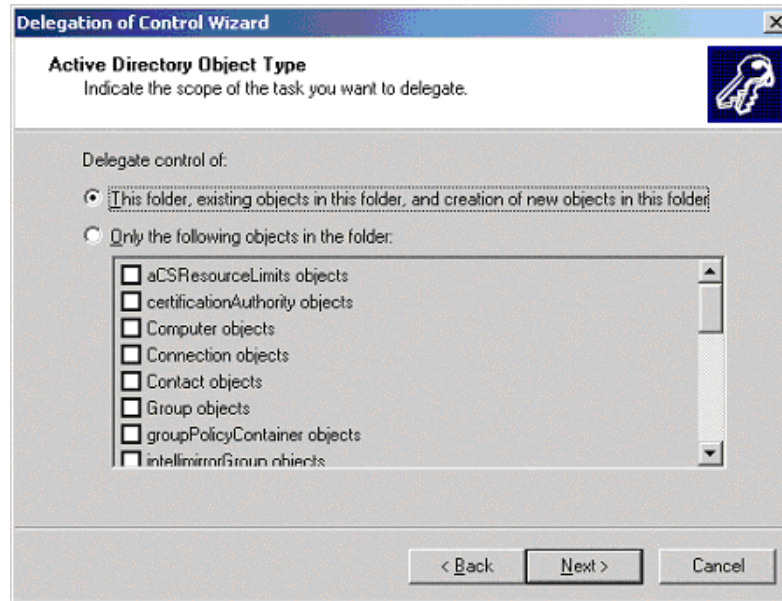


- 4 (Conditional) Select Create a Custom Task to Delegate, then click Next.

If you selected the predefined Users container, skip this step. The following screen won't appear. However, if you selected a container that you created (for example, RDlab), the following screen appears.



- 5 In the Active Directory Object Type window, select This Folder, Existing Objects in This Folder, and Creation of New Objects in This Folder, then click Next.

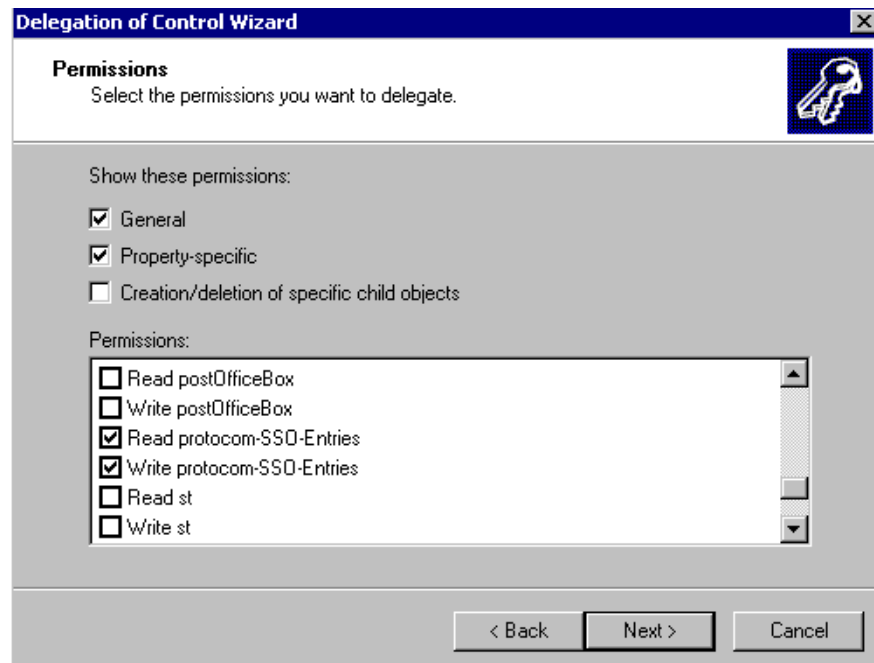


6 Set rights (permissions) on new schema attributes.

6a Under Show These Permissions, check the General and Property-Specific check boxes, click Next, then click Finish.

6b In the Permissions list box, check the Read and Write check boxes for the protocom-SSO-Entries attribute.

The following figure illustrates these check boxes:



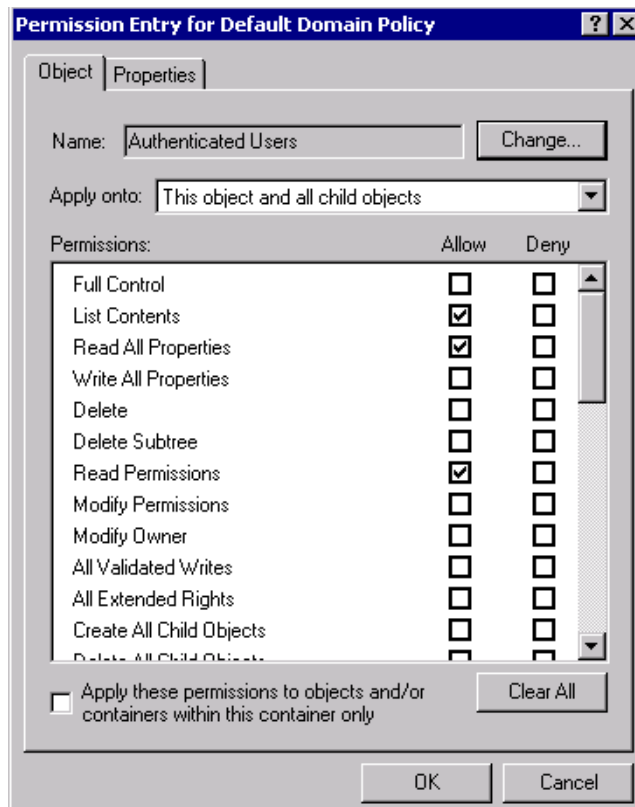
7 Click Next, then click Finish.

Setting the Default Domain Policy

At the domain level, make sure that the Default Domain policy allows all authenticated users to have Read rights to All Properties.

- 1 Expand Active Directory Users and Computers, right-click the domain name, then select Properties.
- 2 Select the Group Policy tab, click Properties, then select the Security tab.
- 3 Click Advanced.
- 4 Select Authentication Users Special, then click View/Edit.
- 5 Under the Allow column, check the Read All Properties check box.

The following figure illustrates this check box:



- 6 Click OK.

Setting Up SecureLogin for NT 4 Domains

To set up SecureLogin for NT 4 domains:

- ♦ Set up a shared folder
- ♦ Administer a corporate script

Setting Up a Shared Folder

- 1 On the NT 4 Domain server, create a shared folder on your shared drive.

Each user must have access to this shared folder. Users' workstations read the corporate script file in this shared folder.

You (the administrator) can assign this folder any name (for example, userdata). This folder handles all prebuilt scripts.

2 On the server, create a home directory for each user.

Every user who logs in to an NT 4 domain must access a home directory created on the server.

From the administrative server, click Start > Administrative Tools > User Manager for Domains, then click New User.

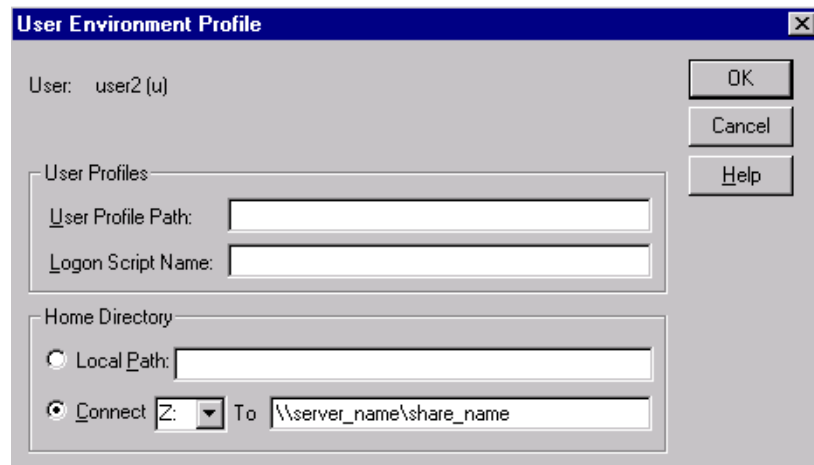
3 Type a name (for example, mkurz).

4 Assign a password.

5 Under Profile, map the home directory to a drive.

You must map a home directory to *drive:\servername\sharename*. In the Connect field, enter the drive letter. In the To field, enter a sharename, which is the IP address, NetBIOS name, or DNS name that distinguishes the server.

The following figure illustrates these fields.

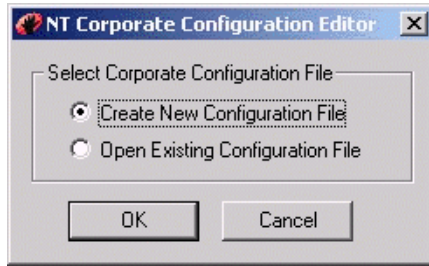


The client logs in to the domain and maps a drive (for example, Z) to the home directory (for example, mkurz). The domain automatically maps the users's home directory. The user's data goes into this directory. When the user logs in, the user has rights to the area.

Administering Scripts for NT

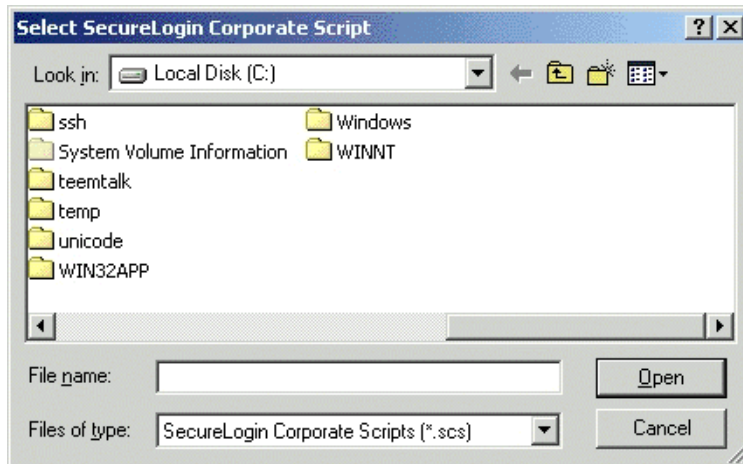
To administer corporate scripts in an NT environment:

- 1** Run the NT Corporate Script Editor from the Start menu.
- 2** From the NT Corporate Configuration Editor screen, select an option.



To create a new corporate script for your environment, select Create New Configuration File. To modify an existing corporate script for your environment, select Open Existing Corporate Script.

- 3 Select a current corporate script or define a name for a new corporate script.



This corporate script (for example, CORP_SCRIPTS.FCS) must be in a shared directory that users have access to. Access to the corporate script files is governed by the standard NT file permissions.

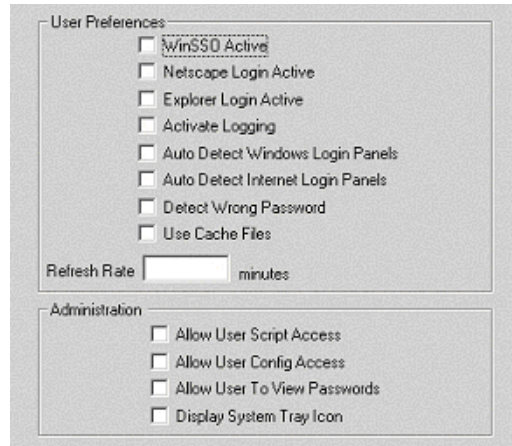
IMPORTANT: Ensure that users have Read Only access to this file.

- 4 Select a configuration to edit.



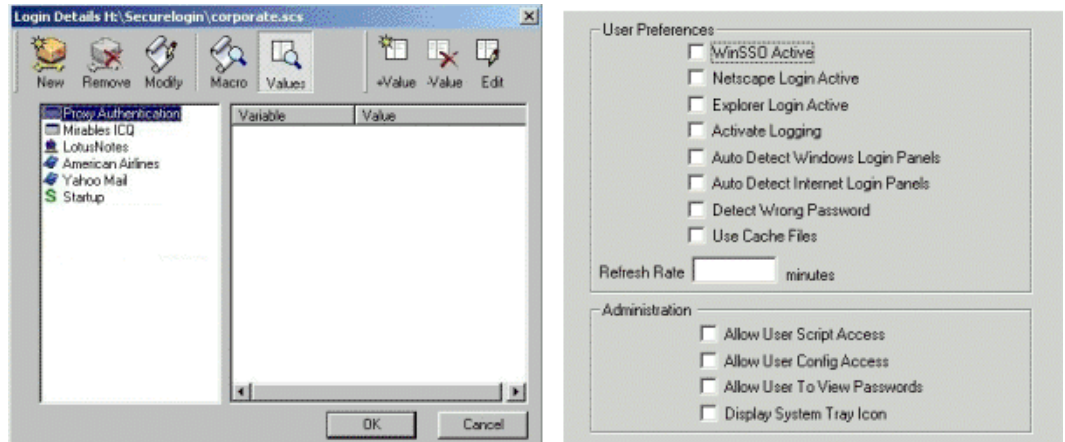
To edit the current corporate scripts, select Edit Scripts.

To edit the current preferences for this corporate script, select Edit Preferences. The following figure illustrates the Preferences property page:



To browse for another script to edit, select Edit Another Script.

The following figure illustrates the main NT script editor window:



This editor operates in the same manner as the user script editor. However, no user variables are stored at this location and there is no Display Passwords option.

To set up a user's configuration to load a corporate script, do one of the following:

- ◆ Pass in the path to the script as the argument to proto.exe.

Edit the registry and change the setting HK_CURRENT_USER/Software/Microsoft/Windows/Current/Run so it has the corporate script location after proto.exe.

- ◆ Remove the above key and create a shortcut similar to the following:

c:\program files\novell\securelogin\proto.exe s:securelogin\production.scs

Place this shortcut in the user's startup folder.

To ensure that SecureLogin will also read from this location, another registry setting exists to keep track of the last location a corporate script was run from. The location for this registry setting is

HKEY_CURRENT_USER\Software\Protocom\SecureLogin\LastNTCORPLocation.

If you close and reload SecureLogin, it will load the corporate script from the last known location.

Installing SecureLogin

To install SecureLogin, run setup.exe at a workstation.

Setup.exe is a multi-lingual installation program. Setup.exe prompts you to select an installation for an English, Brazilian Portuguese, French, German or Spanish system.

Before you log in, make sure that you are authenticated to an NDS or eDirectory tree.

In Step 5, you select an installation option. If you are authenticated to an eDirectory tree and select to install SecureLogin along with SecretStore, installation proceeds as expected. If you are not authenticated, the following scenario occurs.

Scenario—Unusable Login Prompt. You are not authenticated to an eDirectory tree. You select to install the SecureLogin client along with SecretStore. During the installation, you select default settings. The installation program prompts you for a username and password. However, the username field cannot be edited, and no password has been set.

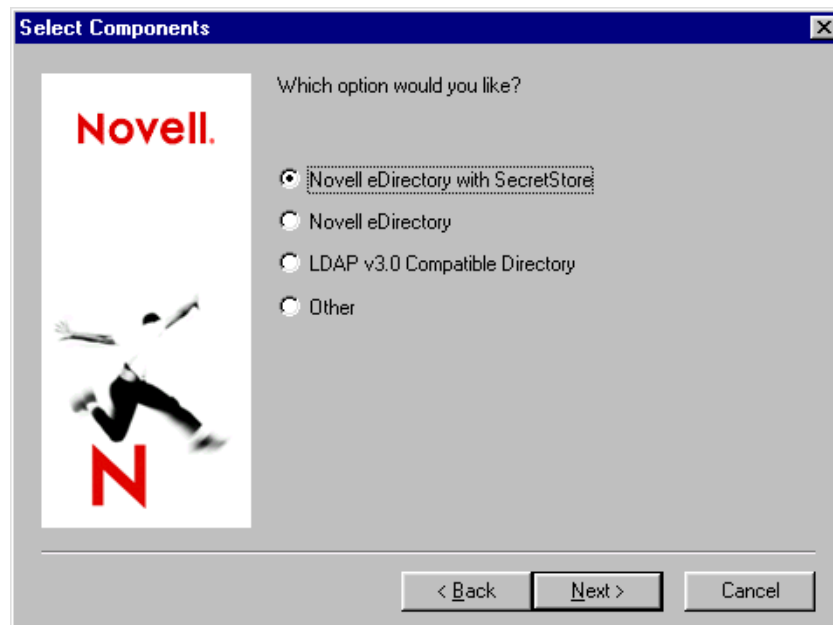
Until you authenticate to eDirectory and set a passphrase or password, SecureLogin continues to display this unusable prompt each time that SecureLogin is started.

- 1 From the SecureLogin CD's securelogin\client directory, run setup.exe.

If autorun is supported on your workstation, the SecureLogin client installation program is launched after you insert the CD. The program is launched according to your regional locale setting. If the default selection doesn't offer the correct language option, cancel the installation and run the version you need.

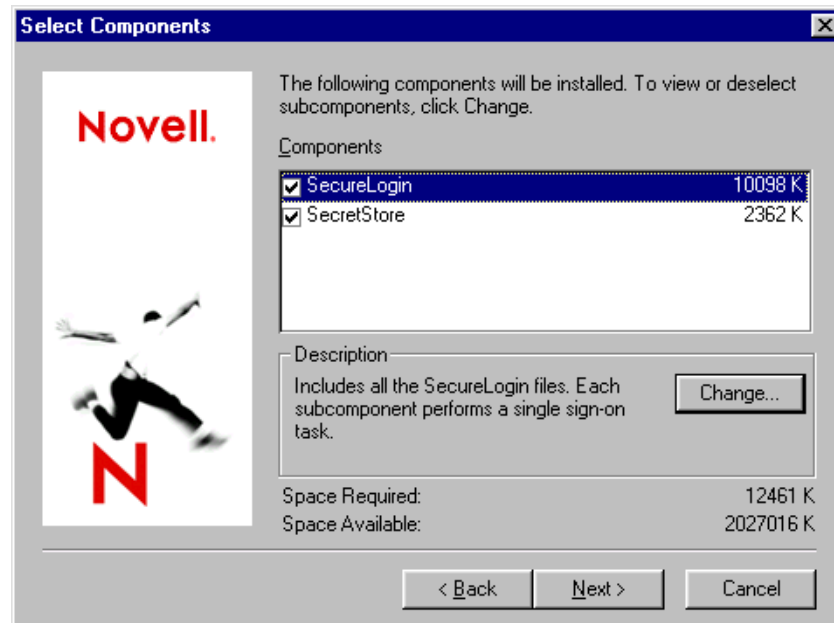
Select the appropriate version from the \SecureLogin\Client directory.

- 2 At the Installing SecureLogin window, click Next and accept the license agreement.
- 3 Select a destination folder, then click Next.
- 4 At the Select Components window, select an option, then click Next.

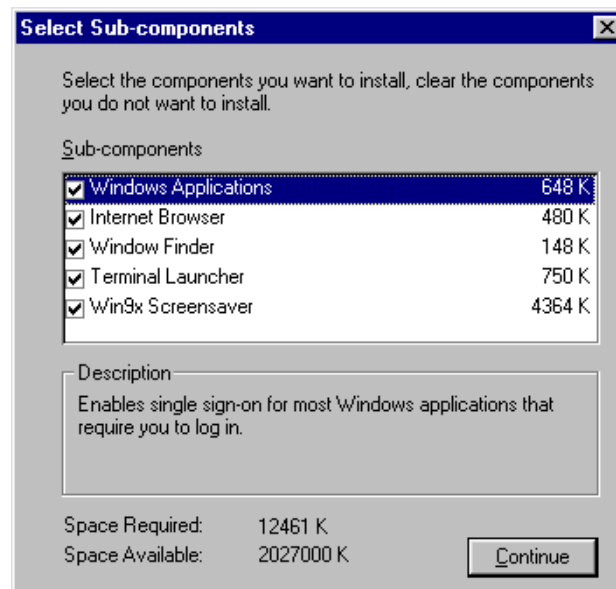


- 5 Select components to install, then click Next.

If you select the Novell eDirectory with SecretStore option, the installation program installs SecureLogin components and SecretStore components by default.



To select subcomponents for an option, click Change, select subcomponents, then click Continue. The following figure illustrates subcomponents that you can select:



The Description panel provides information about a component or subcomponent that you select.

For more information on Terminal Launcher, see [Chapter 8, “Setting Up Terminal Emulation,”](#) on page 117.

NOTE: Concerning the Win9x Screensaver subcomponent, if you are connected to NDS or eDirectory, the password for the screen saver is your NDS password.

If you are not connected to NDS or eDirectory, SecureLogin calls the Windows password system. If you haven't defined a Windows password for the screen saver, through desktop properties, SecureLogin simply unlocks.

The administration tools (snap-ins to ConsoleOne and a corporate script editor) are installed separately. See [“Installing Administration Tools” on page 49](#).

6 Select Post-Install options and click Finish.

6a (Optional) Start SecureLogin now.

If you check this check box, SecureLogin will be active as soon as you complete the installation. You don't need to restart the workstation.

You can access SecureLogin through the SecureLogin icon on the system tray or from Start > Programs > Novell SecureLogin > Novell SecureLogin.

6b (Optional) Start SecureLogin whenever Windows starts up.

If the Program Conflict message appears, see [“Program Conflict” on page 155](#).

7 (Conditional) If you selected the LDAP option, install eDirectory snap-ins.

The snap-ins are located on the Novell SecureLogin CD in the ConsoleOne\Snapins directory. Run edir_snp.zip. Extract files to the consoleone\1.2 directory.

You will find these snap-ins useful if both the following are true:

- ◆ You are installing ConsoleOne 1.3.x for the first time (or you don't currently have these snap-ins).
- ◆ You plan to administer LDAP from the same ConsoleOne that you use to administer Novell SecureLogin and Novell SecretStore.

You can determine whether the eDirectory snap-ins are installed:

- 1** Bring up ConsoleOne 1.3.x.
- 2** Select Help, then click About Snapins.
- 3** Locate the Novell LDAP Snapins entry.

Granting Rights

For eDirectory, rights are automatically granted to a User object if the Novell SecureLogin snap-in to ConsoleOne is loaded when you create that object. If the snap-in isn't loaded when you create the User object, you must grant rights to the User object and to the local workstation cache directory.

Also, if you aren't using eDirectory, you must grant rights to User objects. LDAPSchema.exe doesn't grant rights.

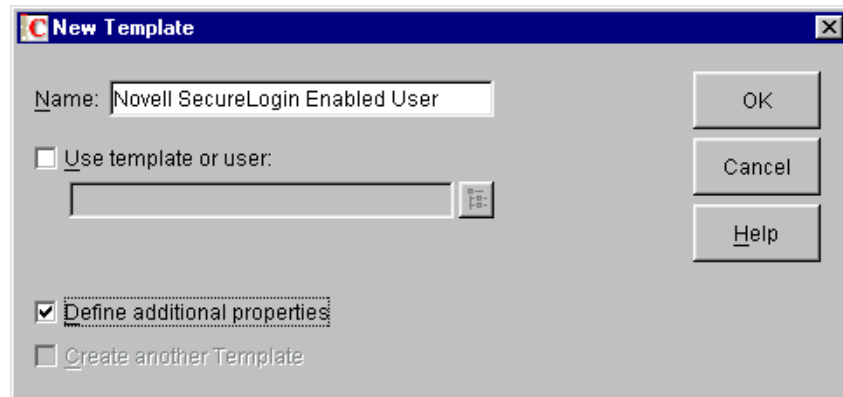
Granting Rights to User Objects

You can automatically or manually grant rights to User objects created after Novell SecureLogin is installed.

Automatically Granting Rights

Using ConsoleOne, you can create a user template that automatically grants rights to required attributes.

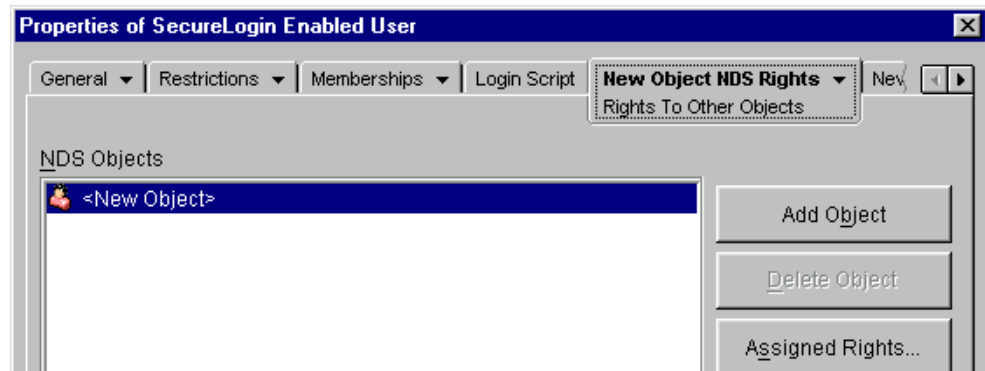
- 1** Select an O or OU Container object that will contain the Template object.
- 2** Create a new object of the class Template.
- 3** At the New Template dialog box, name the template, check the Define Additional Properties check box, then click OK.



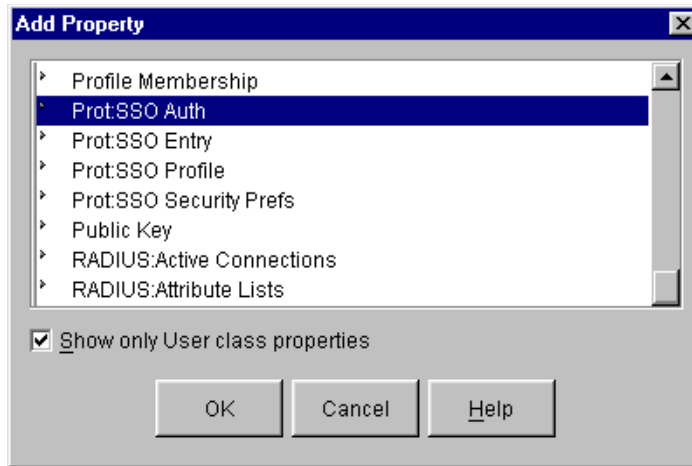
- 4** At the properties page for the new Template object, navigate to and select New Object NDS Rights, then select Rights To Other Objects from the drop-down list.



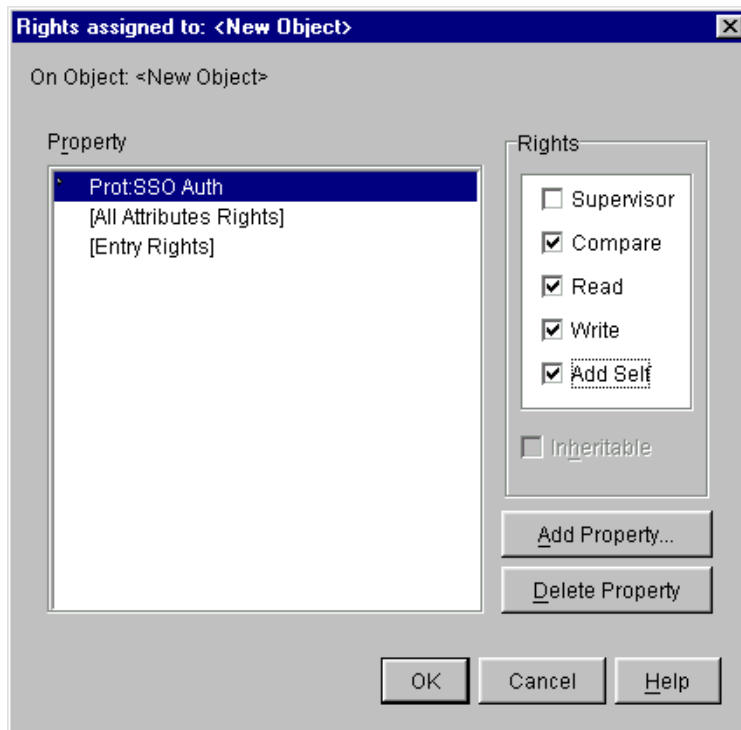
- 5** Click New Object > Assigned Rights.



- 6** Click Add Property, select the Prot:SSO Auth attribute, then click OK.



- 7 At the Rights Assigned To dialog box, check the Compare, Read, Write, and Add Self check boxes, then click OK.



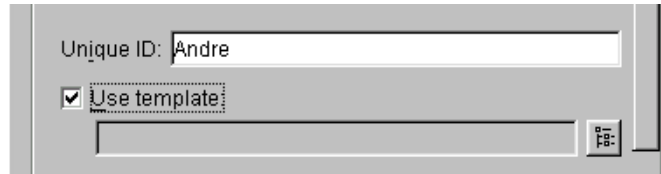
- 8 Configure the Prot:SSO Entry attribute by repeating Steps 5, 6 and 7 for the Prot:SSO Entry attribute.

NOTE: Do not add the Prot:SSO Profile attribute.

- 9 Exit by clicking OK.

To use the new template:

- 1 Create a new User object.
- 2 At the New User property page, enter a name, enter a surname, check the Use Template check box, then click the Browse button.



- 3 Navigate to and select the Template object that you created, then click OK > OK.
- 4 Type and confirm a password for the new user, then click Set Password.

Manually Granting Rights

You can manually grant rights to users created after Novell SecureLogin has been installed. Run NDSSchema.exe, which is typically located in the c:\program files\novell\securelogin directory.

This program extends the schema and grants rights to existing users listed in the installation. No harm is done if the schema has already been extended.

Granting Rights to Local Cache Directories

Users on Windows NT, Windows 2000, and Windows XP must have workstation rights to their local cache directory location. To grant rights, do one of the following:

- ♦ grant rights to the user's cache directory (c:\Program Files\novell\securelogin\cache\VLSC\username)
- ♦ Use the registry setting to relocate the user's cache to a location that he or she has rights to (for example, the user's documents folder).

You can change where the users SecureLogin cache file is located by setting the following registry key:

```
HKEY_LOCAL_MACHINES\SOFTWARE\Protocom\SecureLogin  
"CacheDirectory"="[drive]:\[path]"
```

For example, enter "CacheDirectory"="c:\Documents and Settings\markus".

Installing Administration Tools

Administration tools include the following:

- ♦ ConsoleOne
- ♦ The SecureLogin snap-in to ConsoleOne
- ♦ NDSSchema.exe
- ♦ Loginwatch.exe

To install ConsoleOne on the workstation, run consoleone\c1.exe.

To install the SecureLogin snap-in for ConsoleOne, run consoleone\snapins\NSLSnapin.exe. Select a directory for the snap-in. By default, the installation program installs ConsoleOne files to c:\novell\consoleone\1.2. You can also use this option to install the snap-in file to a server.

You can also install the SecretStore snap-in for ConsoleOne (SSSnapin.exe) from this location.

Other administration tools are in the \securelogin\tools directory.

Running SecureLogin Administration Tools with Language Resources

To run the SecureLogin administration tools (for example, NDSSchema.exe and loginwatch.exe) in one of the supported languages, do one of the following:

- ◆ Copy the tools from \securelogin\tools to the installed SecureLogin program files directory.
For example, copy the tools to c:\Program Files\novell\securelogin.
- ◆ Replace the provided English version of the language resource file (localhero.dll) with one of the language versions found in the software image (\securelogin\client\program files\novell\securelogin\localhero_xxx).

3

Migrating from Earlier Versions

This section contains information about the following:

- ◆ “The Need for Shared Secrets” on page 49
- ◆ “Requirements” on page 49
- ◆ “How the Conversion Tool Works” on page 50
- ◆ “Selecting a Migration Option” on page 50
- ◆ “Converting Novell Single Sign-On 2.1” on page 52

The Need for Shared Secrets

Novell® Single Sign-on (NSSO) 2.1 included Novell SecretStore™ 2.1 and v-GO for Novell Single Sign-on 2.1. You can migrate or remove NSSO 2.1 secrets by using the Novell SecretStore 2.1 Conversion Tool (ss21cvt.exe).

NSSO 2.1 uses SecretStore to securely store credentials (for example, usernames and passwords). Other solutions from Novell, such as Novell DirXML™, also use SecretStore. Before SecretStore 3.0, each application stored information in SecretStore in its own way. Therefore, these credentials could not be shared.

For example, if both Novell DirXML and NSSO 2.1 were deployed in an organization, users in that organization provided their usernames and passwords twice, first in NSSO 2.1 and then in DirXML. In addition, if the credentials were ever changed, they had to be changed in both applications.

To resolve these issues, a Shared Secret Format Recommendation was released with SecretStore 3.0. The recommendation allows for applications that use SecretStore to share application credentials, thus relieving the administrative burden of storing duplicate passwords and of synchronizing passwords among applications.

Any single sign-on enabling application that conforms to the Shared Secret Format Recommendation can leverage these secrets. Novell SecureLogin 3.0 conforms to this recommendation. NSSO 2.1 used a different format. ss21cvt.exe converts NSSO 2.1 secrets to the shared secret format.

For more information on shared secrets, see [Sharing Secrets](#) in the *Novell SecretStore Administration Guide*.

Requirements

Workstation

- ◆ Novell Client SecretStore 3.0

- ◆ Novell International Cryptographic Infrastructure 2.0.2

Server

Have Novell SecretStore 3.0 running on the server.

How the Conversion Tool Works

By default, the conversion tool performs the conversion process for the eDirectory (or earlier versions of NDS) tree that is specified by the primary connection. You can use either of the following options:

- ◆ Before running the conversion tool, change the primary connection to the tree and server that require the conversion.
- ◆ Pass in a command line parameter. See [“Converting Novell Single Sign-On 2.1” on page 52](#).

Only individual users can read their own secrets. Therefore, it is not possible for the conversion tool to convert secrets for a user that is different than the authenticated user.

So that the conversion tool can map an NSSO 2.1 secret to a shared secret, the tool must know which application the 2.1 secret was created for. Not all of the information needed for mapping the secrets is contained in every NSSO 2.1 secret. Whether or not enough information exists depends on the type of application that the NSSO 2.1 secret exists for.

For example, all of the necessary information exists for Web and user-defined applications. However, not all of the necessary information exists for pre-defined and admin-defined applications. This tool does not convert secrets for mainframe applications.

For the pre-defined and admin-defined applications, the information is contained in two files. The information needed to map the secrets for predefined applications is in the `applist.ini` file, which is located where NSSO 2.1 is installed. If this file is not found, pre-defined secrets will not be converted.

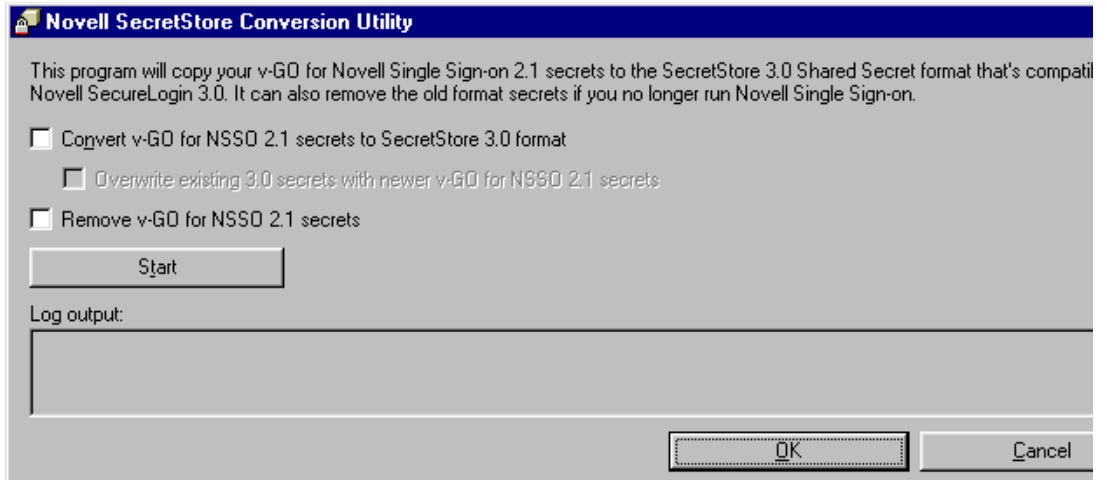
Information needed to map the secrets for administrator-defined applications is in the `entlist.ini` file. If the administrator has defined any applications in the directory, `ENTLIST.INI` is created when NSSO 2.1 starts. If `entlist.ini` is not found, administrator-defined secrets will not be converted.

The conversion tool searches for `applist.ini` and `entlist.ini` in the same directory that the conversion tool is in. Thus it is possible to deploy these files along with this tool if NSSO 2.1 is not currently installed. If the conversion tool doesn't find `applist.ini`, the tool searches for it where NSSO 2.1 is installed in the Passlogix folder. If the conversion tool doesn't find `entlist.ini`, the tool searches for it in the Windows directory (or the WINNT directory on NT workstations).

After mapping the secrets, the conversion tool converts the data by extracting the usernames or passwords (or both) from the NSSO 2.1 secrets. The tool then populates corresponding shared secrets with the usernames and passwords.

Selecting a Migration Option

You can migrate secrets by using one or more of the options illustrated in the following figure:



You can convert secrets before or after installing SecureLogin 3.0. However, we recommend that you convert before uninstalling Novell Single Sign-on 2.1. Entlist.ini and applist.ini must be in specific locations so that ss21cvt.exe can locate these files. If you run ss21cvt.exe before you uninstall NSSO 2.1, you don't have to manually place the two .ini files.

Overwriting Secrets

The first option is to convert v-GO for NSSO 2.1 secrets to SecretStore 3.0 format. To use this option, check the Convert check box, but don't check the Overwrite check box.

If you don't check the Overwrite check box, and if a 3.0 secrets exists, NSSO 2.1 secrets don't overwrite SecretStore 3.0 secrets.

Scenario: Convert is Enabled but Override is Disabled

Rie is a user at Digital Airlines, which has been using Novell Single Sign-on 2.1. You install SecureLogin 3.0. Rie uses 3.0 to log in to Web sites. As the following table illustrates, secrets for both 2.1 and 3.0 for Rie exist in eDirectory.

Secret and Date	Secret and Date	Results during a Migration
2.1a, created October 1	3.0 a, created September 1	No change
2.1b, created October 1	3.0 b, created September 1	No change
2.1c, created October 1		2.1c becomes 3.0c

You run the conversion program. Because secrets 3.0a and 3.0b exist, Rie's secrets 2.1a and 2.1b are not converted to 3.0 format. Secret 3.0c does not exist. Therefore, the conversion program converts 2.1c to 3.0 format and writes it as 3.0c.

Converting and Overwriting Secrets

The second option is to convert NSSO 2.1 secrets and overwrite SecretStore 3.0 secrets. To use this option, check both the Convert check box and the Overwrite check box.

You can't overwrite secrets without converting them. If you enable Overwrite, the conversion program only overwrites secrets that have earlier dates. The 2.1 secrets that have a later date than 3.0 secrets overwrite the older 3.0 secrets during the conversion process.

Scenario: Convert and Overwrite Are Both Enabled

Claire works at the VMP company, which has been using Novell Single Sign-on 2.1. You install SecureLogin 3.0. Claire uses 3.0 to log in to Web sites. As the following table illustrates, secrets for both 2.1 and 3.0 for Claire exist.

You run the conversion program. Because Secret 2.1a is newer than Secret 3.0a, the program converts Secret 2.1a to SecretStore 3.0 format and overwrites 3.0a.

Secret and Date	Secret and Date	Result during a Migration
2.1a, created October 1	3.0a, created September 1	3.0a is overwritten
2.1b, created October 1	3.0b, created November 1	2.1b is not converted
2.1c, created October 1		2.1c becomes 3.0c

Secret 2.1a is older than Secret 3.0b. The program does not convert 2.1b to 3.0 format. Secret 3.0b is not overwritten.

Because 3.0c does not exist, the conversion program converts 2.1c to 3.0 format and writes it as 3.0c.

Removing Secrets

The third option is to remove NSSO 2.1 secrets. To do this, check the Remove check box.

If you select both 'Convert and Remove, the conversion program first tries to convert 2.1 secrets to 3.0 secrets. It then removes the 2.1 secrets.

If you select Remove but do not select Convert, the conversion program only removes Novell Single Sign-on 2.1 secrets. It does not delete connector secrets or secrets owned by other services.

Converting Novell Single Sign-On 2.1

You can convert from NSSO 2.1 before or after installing SecureLogin 3.0.

- 1 From the securelogin\tools directory, run `ss21cvt.exe`.
- 2 Select an option, then click OK.

You can run the conversion options from the command line:

- ◆ To convert, enter `ss21cvt -convert`

This option sets the tool to convert Single Sign-on 2.1 secrets to the shared secret format.

- ◆ To convert and overwrite, enter `ss21cvt -convert -overwrite`

This option sets the tool to convert Single Sign-on 2.1 secrets to the shared secret format and overwrite existing shared secrets, provided the corresponding Single Sign-on 2.1 secrets are newer.

- ◆ To remove, enter `ss21cvt -remove`

This option sets the tool to remove Single Sign-on 2.1 secrets. If you specify to both convert and remove Single Sign-on 2.1 secrets, the conversion occurs before any secrets are removed.

- ◆ To write conversion information to a log file, enter **ss21cvt -log [path] [filename]**

For example, enter `-log c:\Novell\upgr21.txt`.

This option is useful if you run the tool in suppressed mode but still want to know the details of what the tool did. If the file already exists, the file will be appended to instead of truncated. If any spaces exist in the path, you must enclose the path with quotation marks (“”).

- ◆ To run the conversion program without displaying information, enter **ss21cvt -suppress**.

This option instructs the tool to not display the dialog, but instead to automatically perform the tasks that were specified on the command line. Most likely, you will use ZENworks® to convert secrets on users’ workstations, and include `-suppress` in the script.

- ◆ To run the conversion tool for a specified eDirectory tree, enter **ss21cvt -tree [treename]**.

If the tree is not found, the tool attempts to perform the tasks against the primary connection. If any spaces exist in the name, you must enclose the name with quotation marks (“”).

Migrating from Novell SecureLogin 2.5

Because Novell SecureLogin 2.5 didn’t use SecretStore, you don’t have to run the conversion utility when you upgrade to SecureLogin 3.0. SecureLogin 3.0 automatically upgrades a 2.5 cache and upgrades 2.5 directory attributes.

Installers could deploy SecureLogin 2.5 to run in standalone mode or with eDirectory. Even if SecureLogin 2.5 was deployed to work with eDirectory, a cache most likely exists on the workstation, unless the administrator turned that capability off. SecureLogin 3.0 recognizes the cache left by SecureLogin 2.5 and automatically works with it.

If all SecureLogin 2.5 data was stored in eDirectory, and if SecureLogin 3.0 is installed to work with SecretStore, SecureLogin 3.0 is able to use the data from SecureLogin 2.5. Usage occurs because SecureLogin 3.0 still uses the Prot:* attributes in the directory, even if deployed to use SecretStore.

To migrate from SecureLogin 2.5:

- 1** While running `setup.exe`, select to uninstall SecureLogin 2.5 from the workstation.
- 2** Install SecureLogin 3.0.

4

Adding Applications for Single Sign-On

After SecureLogin is installed on your desktop, SecureLogin watches for new applications. Upon detecting a new application that requires a login, SecureLogin prompts you to use SecureLogin wizards that enable those applications for single sign-on.

The Add Applications wizard enables you to single sign-on to Windows applications and Web pages. The Terminal Launcher wizard enables you to single-sign-on to mainframe applications. To enable applications for terminal emulators, see [Configuration Guide for Terminal Emulators](#).

These wizards help you generate login scripts and store variables. The scripts avoid lengthy setup periods.

This section contains information on the following:

- ◆ [“Encountering New-Application Prompts” on page 55](#)
- ◆ [“Adding and Enabling Applications” on page 56](#)

Encountering New-Application Prompts

Prompts for Windows Applications

After detecting a Windows* login panel, SecureLogin displays the following dialog box. (If the application has a prebuilt script, the initial text varies.)

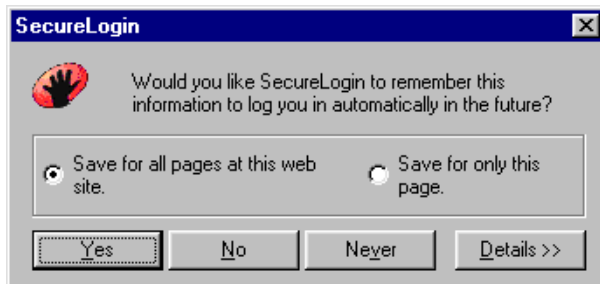


You can choose not to enable single sign-on for this application. If you select not to be prompted during future encounters, SecureLogin generates a blank script for this application and doesn't prompt you again.

If you decide to set up single sign-on, SecureLogin launches the Add Applications wizard, which guides you through the setup. The next time you open the application, SecureLogin authenticates for you.

Prompts for Web Pages

When you navigate to a Web page that has a login field and then submit your login information, SecureLogin prompts you whether to enable single sign-on for the site, for only the current page, or not at all.



If you select Yes, SecureLogin extracts the login information from the Web page and stores it for future single sign-on authentication.

Adding and Enabling Applications

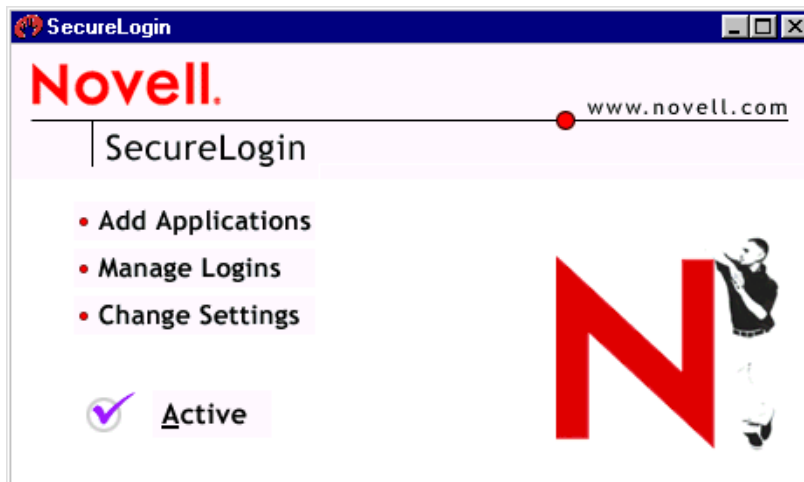
The Add Application wizard enables you to easily capture login data for Windows applications and Web pages. Many applications have prebuilt scripts.

Using Prebuilt Scripts

To add an application that has a prebuilt script:

- 1 Right-click the SecureLogin icon on the system tray, then click Add Applications.

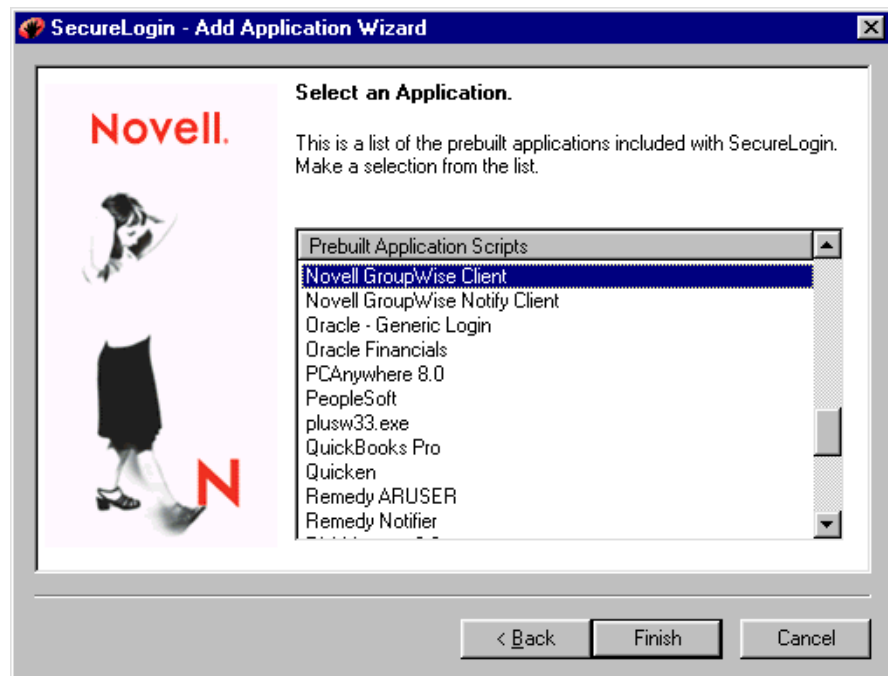
You can also click Start > Programs > Novell SecureLogin > Novell SecureLogin > Add Applications.



- 2 At the Welcome window, click Next.
- 3 Click Prebuilt Scripts, then click Next.



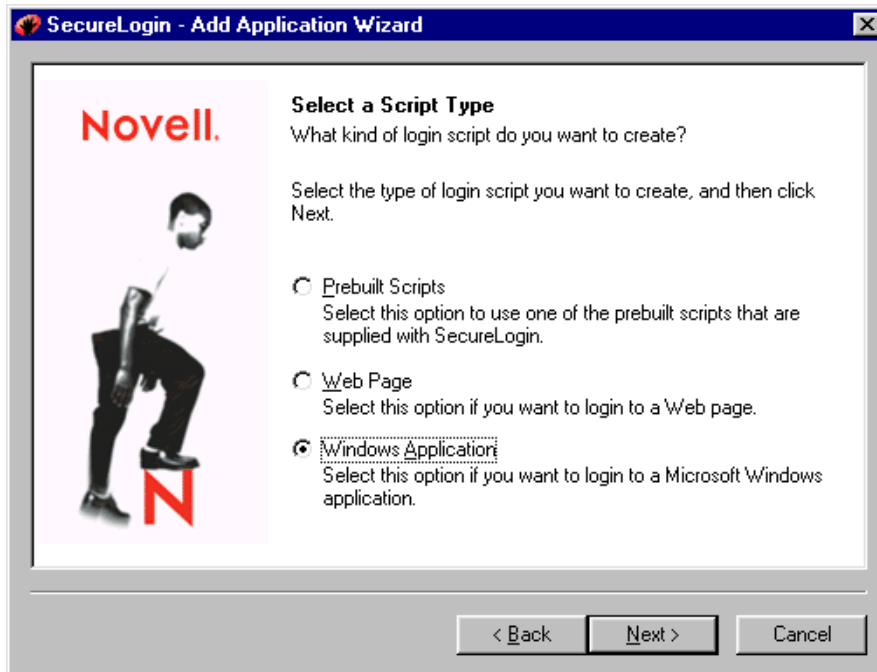
- 4 Select the application from the list, then click Finish.



After adding the application, create or select a login for it.

Adding a Windows Application

- 1 Launch the application and drag the login panel to one side of the screen.
- 2 Right-click the SecureLogin icon on the system tray, then click Add Applications.
- 3 Click Next > Windows Application > Next.



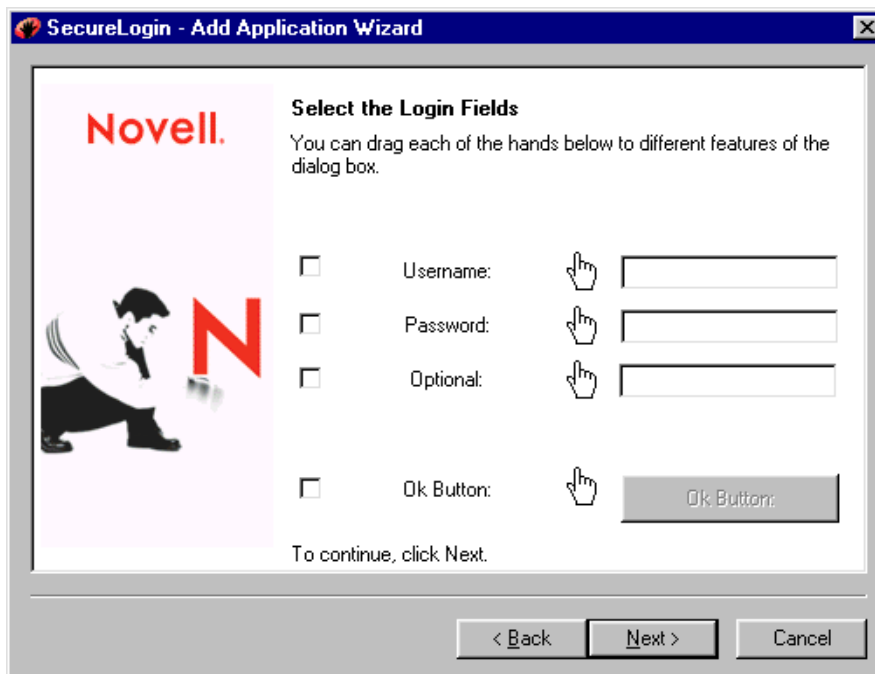
- 4 At the Setup a Windows Application window, review the steps, then click Next.
- 5 Left-click the hand icon in the Add Application wizard, then drag the hand icon to the title bar of the login panel.



- 6 At the Select Window Function window, select Login Window from the drop-down list, then click Next.



7 Select the login fields.



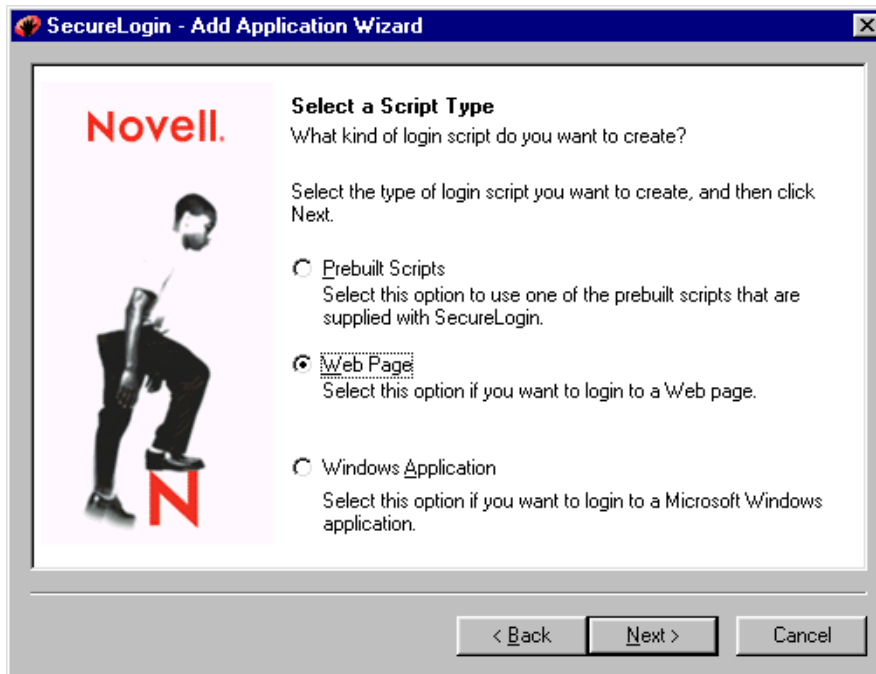
- 7a** Drag the hand icons and drop them onto the Username, Password, and (if appropriate) Optional fields on the login panel
- 7b** Drag the hand icon for the OK Button and drop it onto the OK, Next, Continue, or similar button.
- 7c** Click Next.

- 8 At the Name the Script window, accept the default name (or type a new one) for the application, then click Finish.

After adding the application, create or select a login for it.

Adding a Web Application

- 1 From a browser, go to the URL of the Web page that you want to log in to.
- 2 Right-click the SecureLogin icon on the system tray, then click Add Applications.
- 3 Click Next > Web Page > Next.



- 4 At the Setup a Web Page window, copy and paste the URL, type your username and password, then click Finish.



After adding the application, create or select a login for it.

If you have difficulty logging on to a Web site, see [“Logging in to Difficult Web Sites”](#) on [page 153](#).

5

Managing SecureLogin

SecureLogin captures login information and saves that information to your workstation and Novell® eDirectory™ or earlier versions of NDS®. Using SecureLogin management tools, you can view and edit your saved information.

This section provides information about the following:

- ◆ “Setting Up a Passphrase Question and Answer” on page 63
- ◆ “Setting Passphrase Settings” on page 65
- ◆ “Copying, Exporting, and Importing SecureLogin Settings” on page 68
- ◆ “Managing Logins” on page 70
- ◆ “Customizing SecureLogin” on page 75
- ◆ “Using SecureLogin with Client Software” on page 83
- ◆ “Administering SecureLogin for Active Directory” on page 83
- ◆ “Sharing Secrets” on page 85
- ◆ “Setting Up Multiple Logins for an Application” on page 86
- ◆ “Changing the Startup Order of Applications” on page 87
- ◆ “Using Login Watcher” on page 90

Setting Up a Passphrase Question and Answer

When you use SecureLogin, information for logging in to applications is collected and saved to one or more data stores. This information is used to provide single sign-on to those applications in the future. Depending upon the installation options that you have chosen, login data is stored in the following areas:

- ◆ An encrypted cache file on your workstation
- ◆ Encrypted attributes on your User object in eDirectory
- ◆ Your SecretStore in eDirectory

As an enhanced security feature, SecureLogin detects administrative NDS or eDirectory password changes so that no one else can gain access to your login data stores.

When first run, SecureLogin prompts you for a passphrase question and answer. This information helps you access your login data in the following situations:

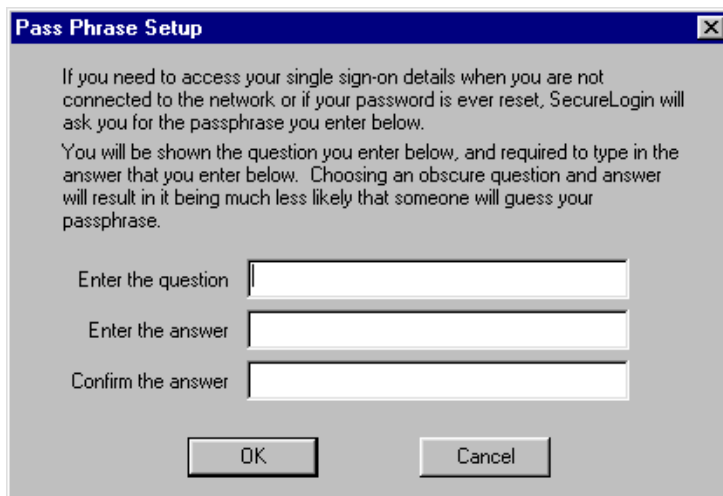
- ◆ You aren’t authenticated to NDS or eDirectory.
- ◆ The network connection is lost.
- ◆ You are using a laptop computer and are out of contact with the network

- ◆ You forget your NDS password and have it reset for you.
- ◆ Your credential store was locked when an administrator inappropriately reset your NDS password.

Choose passphrase information that you'll be able to recall months or years from now.

NOTE: For a passphrase to display properly on multi-byte platforms (for example, Japanese and Chinese), users must use single-byte characters when entering a passphrase.

The following figure illustrates the dialog box that collects your passphrase question and a passphrase password:



After the passphrase information is entered, you won't see this screen again unless you use the Change Passphrase option to reset your password or log in to eDirectory as a different user.

When using SecretStore, a specially-designated administrator might unlock your directory-based data stores on your behalf. Therefore, don't be surprised if a call to the help desk to have your eDirectory password reset doesn't result in a passphrase answer prompt when you next login. This feature is only available when using SecretStore and the SecretStore Administrator feature. (For more information, see [Setting Up a SecretStore Administrator](#) *Novell SecretStore Administration Guide*.)

Disabling the Local Cache

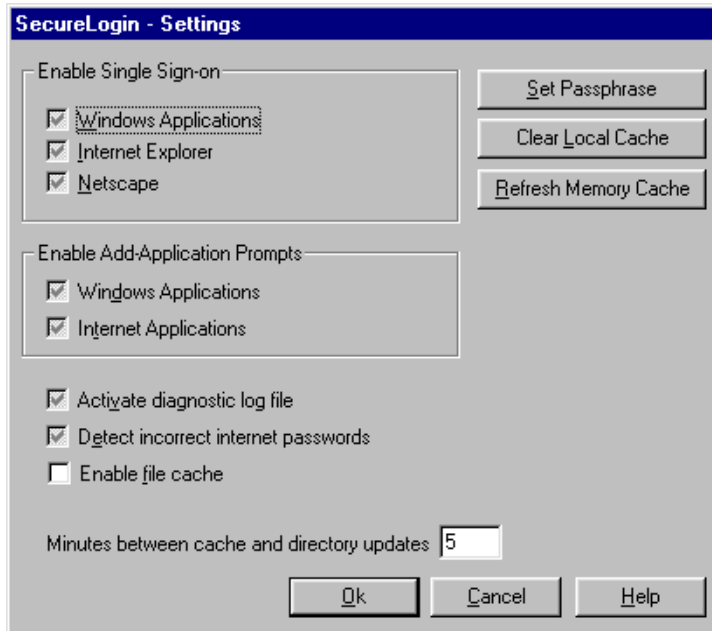
Your local login data can be stored for offline access in encrypted files on your workstation. These cache files are located in the program files\novell\securelogin\cache directory. The files are triple Data Encryption Standard (3DES) encrypted.

If you forget the cache passphrase answer password and are not able to log in using your eDirectory password, you will have to delete and recreate the cache files. SecureLogin automatically recreates cache files, provided you are authenticated to the network.

To turn off caching functionality:

- 1** At the SecureLogin main screen, select Change Settings.
- 2** Uncheck the Enable File Cache check box, then click OK.

The following figure illustrates this check box:



Setting Passphrase Settings

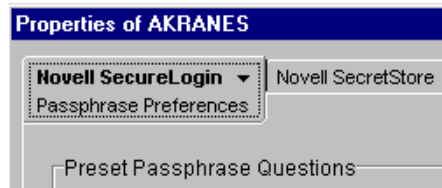
When users first run SecureLogin, SecureLogin asks them to set up a passphrase question and a passphrase answer. SecureLogin uses the question and answer to ensure that no one else uses their login credentials to access their applications.

You can provide customized questions for users to respond to as well as allow users to enter a passphrase question.

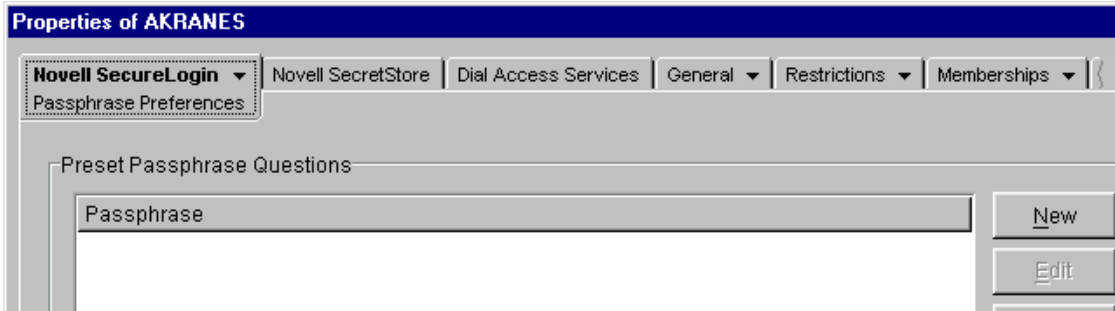
Providing Preset Passphrase Questions

To provide preset passphrase questions for users:

- 1** In ConsoleOne, right-click an object, then click Properties.
- 2** Click Novell SecureLogin, then select Passphrase Preferences from the drop-down list.

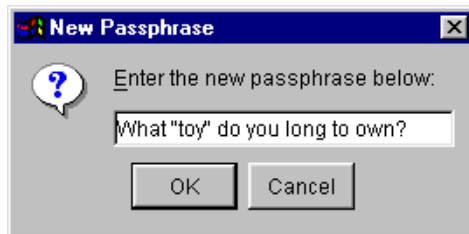


- 3** At the Passphrase dialog box, click New.



To edit a passphrase question, select it, click Edit, make changes, click OK, and then click Apply.

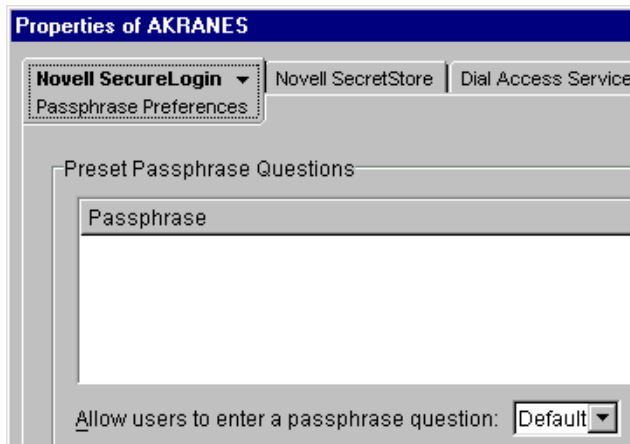
- 4 Type a question in the New Passphrase text box, then click OK.



- 5 Click Apply.

Allowing Users to Enter Passphrase Questions

To allow users to enter their own passphrase questions, select Yes from the Allow Users To Enter a Passphrase Question drop-down list.

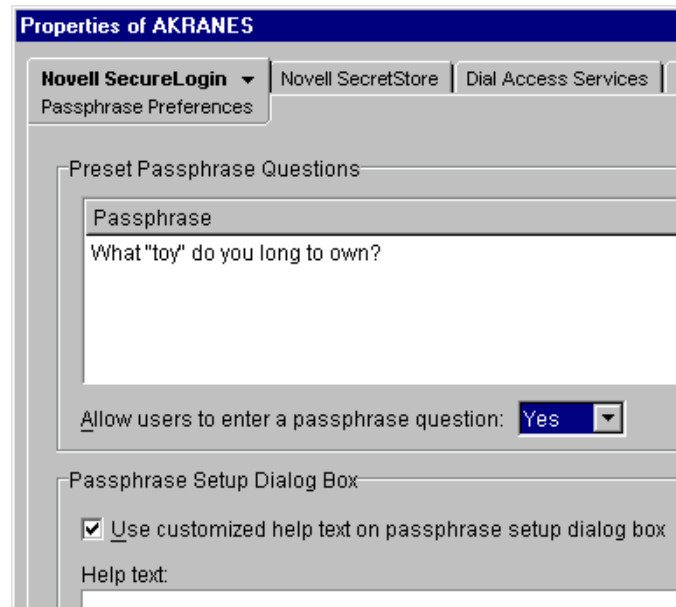


If you select No, users must respond to a predefined question before they can enter or change a passphrase.

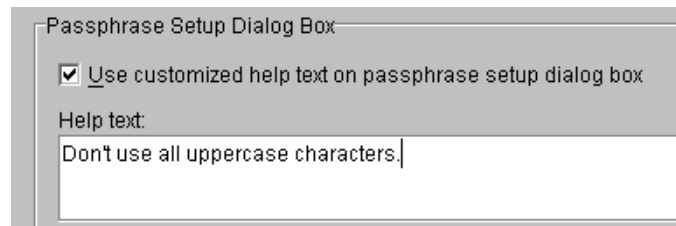
Using a Customized Prompt to Change a Passphrase Question

When users first encounter SecureLogin, SecureLogin provides introductory text that explains passphrases. You can edit that text and provide a customized introduction.

- 1 From the Passphrase Preferences page, go to the Passphrase Setup dialog box.
- 2 Check the Use Customized Help Text text box.



- 3 Type text in the Help Text pane.



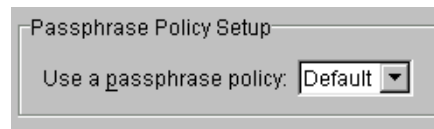
If you uncheck the check box, your customized prompt is disabled.

- 4 Click Apply > OK.

Using a Passphrase Policy

By default, SecureLogin requires a passphrase that has at least six characters. To set other requirements:

- 1 At the Passphrase Preferences page, go to the Passphrase Policy Setup dialog box.



- 2 Select Yes from the drop-down list.
- 3 Click Edit to display parameters for the policy.
- 4 Set parameters, then click OK.

You can also set advanced settings. For more information, see the Help system in ConsoleOne.

Copying, Exporting, and Importing SecureLogin Settings

The Copy Settings feature enables you to copy SecureLogin settings (data) from one object in an NDS or eDirectory tree to one or more objects in that tree. The objects can be in the same context or in a different context. You can't copy settings from one tree to another.

However, you can export or import settings from one tree to a target tree. After settings are exported or imported, you can then copy them from within the target tree.

The Copy feature saves settings internally (RAM) and copies to objects. The Export feature saves the settings externally, to an XML file. You can then use the XML file time and again to import settings to objects.

Copy Settings doesn't copy, export, or import variables. Therefore, usernames and passwords are not copied, exported, or imported.

Copying SecureLogin Settings

To copy SecureLogin settings, use the following guidelines:

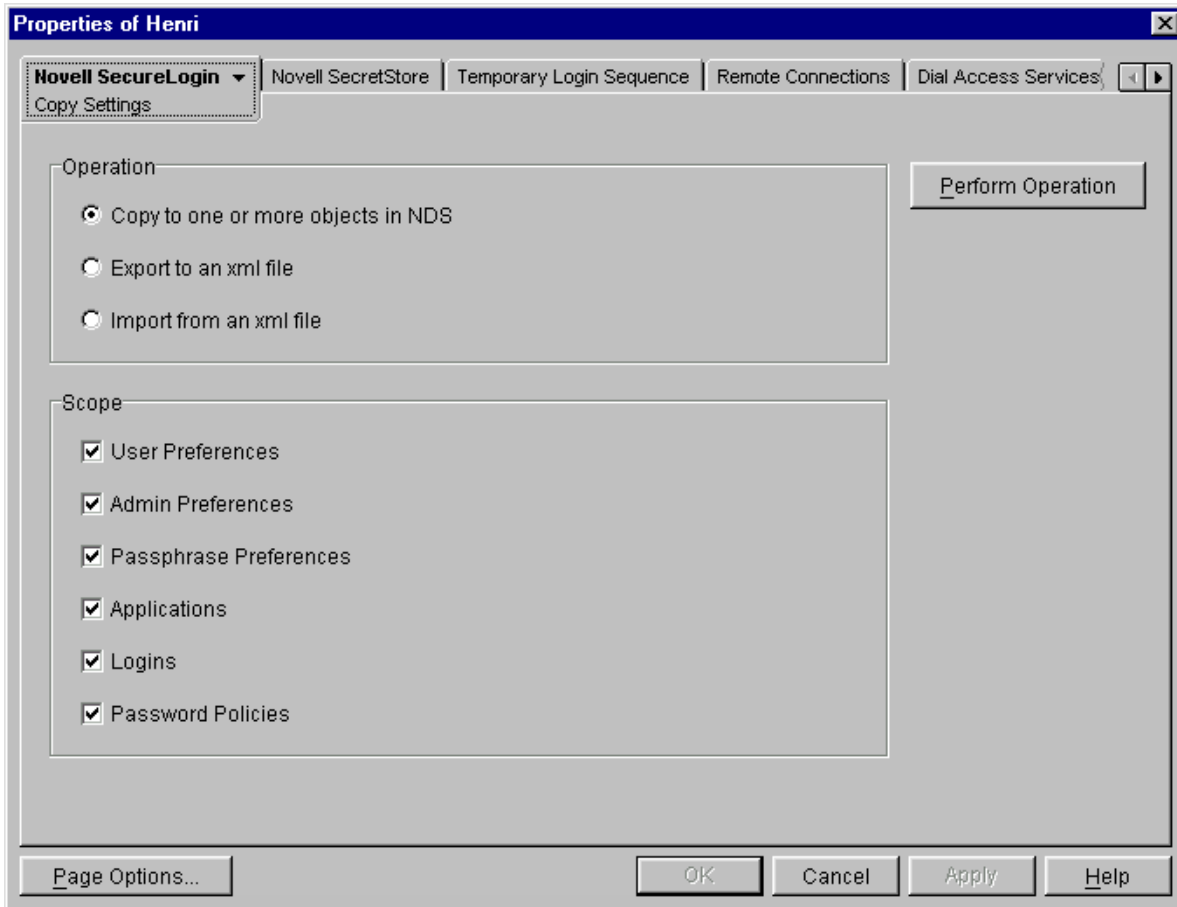
- ◆ Typically, copy from a User object to a User object and from a Container object to a Container object.
- ◆ Copy settings to an object in the same context, a parallel context, or a subordinate context. Don't copy settings from an object in a subordinate context to an object in a superior (higher in the tree) context.
- ◆ Copy items that have local settings. When inheritable settings are copied, they become local settings on the object that the settings are copied to. Such copied settings might have broken login-to-application links.

To copy settings:

- 1** In ConsoleOne[®], right-click the object that has the settings that you want to copy, then click Properties.

You can select an Organization, Organizational Unit, Locality, Country, or User object.

- 2** At the Novell SecureLogin tab, select Copy Settings.



- 3** Select Copy to One or More Objects in NDS, then check all check boxes for settings that you want to copy.

By default, all settings are selected. To limit the scope, uncheck check boxes for settings that you don't want to copy, export, or import.

For example, if you only want to copy User Preferences, uncheck the other check boxes.

- 4** Click Perform Operation.
- 5** At the Select Objects page, select one or more objects that you want to copy the settings to.
You can browse to and select one or more objects from other contexts, but you can't select objects from other trees. To select an object, click it, then click Select. Selected objects appear in the Selected Objects pane.
- 6** Click OK.

Exporting SecureLogin Settings

You can export settings from one tree and import them into the same tree or a different tree. The Export and Import options operate on the same settings as Copy Settings.

To export and import settings, you use .XML files. The files have a corresponding XML schema file (nsldata.xsd).

The XML schema file specifies XML tags and type of data. The file controls how SecureLogin behaves.

To export SecureLogin Settings:

- 1** Right-click the object that has the settings that you want to export, then click Properties.
- 2** At the Novell SecureLogin tab, select Copy Settings.
- 3** Select Export to an xml File, then check all check boxes for settings that you want to export.
To limit the scope, uncheck check boxes for settings that you don't want to export.
- 4** Click Perform Operation.
- 5** Save the settings to an XML file.

Navigate to the directory where you want to save the XML file, specify the filename, then click Save.

The settings are ready to import to another object.

You can edit exported XML files. The XML schema file is provided so that you can verify any modified XML file. However, an easier way to verify a modified file is to import it. The SecureLogin snap-in to ConsoleOne reports an error if the modified file has incorrect syntax or some other problem.

Importing SecureLogin Settings

To import SecureLogin settings:

- 1** Right-click the object that has you want to import the settings to, then click Properties.
- 2** At the Novell SecureLogin tab, select Copy Settings.
- 3** Select Import, then check all check boxes for settings that you want to import.
To limit the scope, uncheck check boxes for settings that you don't want to import.
- 4** Click Perform Operation.
- 5** Navigate to and select the XML file that contains the settings that you want to import, then click Open.

When you import settings from an XML file, SecureLogin validates the XML file against the XML schema. An invalid XML file is rejected.

You can only import settings to one object at a time. However, after importing you can then copy (within the target tree) settings that you imported.

Managing Logins

Two options enable users and you to view, edit, add, or delete login information:

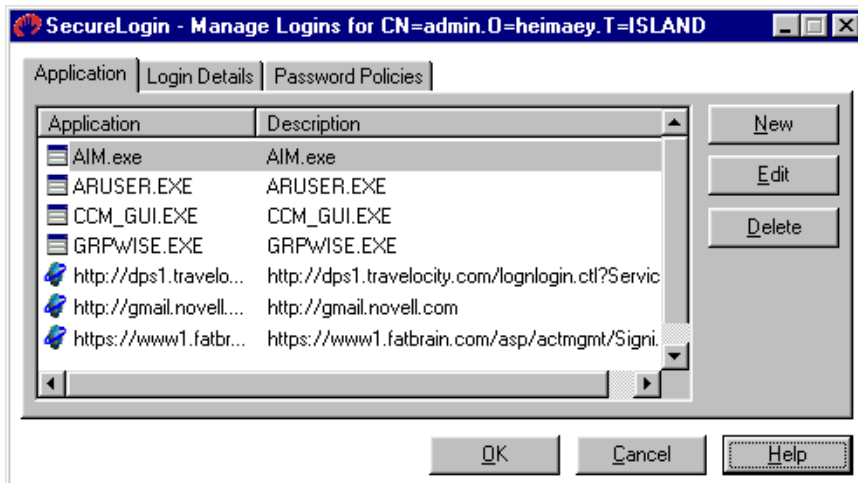
- ◆ The SecureLogin application
- ◆ The Manage Logins property page in ConsoleOne

Managing Logins through the SecureLogin Application

Each workstation running SecureLogin has an administration tool. Users use this tool to manage their single sign-on credentials and change SecureLogin settings. Management tasks include adding applications and looking after login details.

This tool can only alter the current user's SecureLogin information. To create corporate scripts that affect multiple users, you must use the Admin Preferences tab in the ConsoleOne snap-in.

The following figure illustrates the main tabs for the Manage Logins option.



To launch this tool, select Manage Logins from the SecureLogin main screen or right-click the icon on the system tray.

If you make a mistake while adding, editing, or removing details, click Cancel to close the application without saving any changes.

Some applications have prebuilt scripts.

NOTE: Prebuilt application scripts are provided only for English applications. If you deploy SecureLogin in a multi-lingual environment, you might need to modify or create scripts that recognize localized application window titles or text controls.

Enabling Single Sign-On through ConsoleOne

Using ConsoleOne, you can enable single sign-on and manage logins at the User or Container object level.

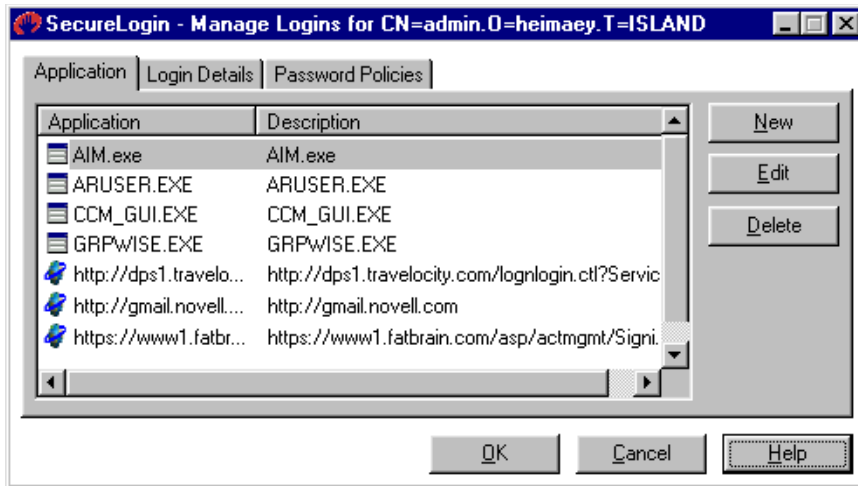
- 1 Select the object, then click Properties.
- 2 Select the Novell SecureLogin tab, then click Manage Logins.

Adding Applications to the List of Logins

Use the Applications page to list applications that will enable users to easily log in through SecureLogin's single sign-on functionality.

The Application Column

The Application column lists Web, Windows, and terminal applications that you have set up for single sign-on. Icons indicate the application type.



Some applications have prebuilt scripts.

NOTE: Prebuilt application scripts are provided only for English applications. If you deploy SecureLogin in a multi-lingual environment, you might need to modify or create scripts that recognize localized application window titles or text controls.

To add one of these applications to the list:

- 1** Click New, then click Select a Prebuilt Application Script.
- 2** Select an application, then click OK.
- 3** Save the list.

To save a setting and continue working, click Apply. To save a setting and exit, click OK.

- 4** Create a login for the application by using the Logins page.
- 5** Link the application and login entry by using the Login Details page.
 - 5a** Select the application, click Edit, then click New.
 - 5b** Select an existing login or create one.

To add applications that don't have prebuilt scripts:

- 1** At the Applications page, click New.
- 2** Select the New Application option.
- 3** Type a name.
- 4** Select a type (Startup, Terminal Launcher, Web, Windows), then click OK.
- 5** Save the list.

To save a setting and continue working, click Apply. To save a setting and exit, click OK.

- 6** Create a login for the application by using the Logins page.
- 7** Link the application and login entry by using the Login Details page.
 - 7a** Select the application, click Edit, then click New.
 - 7b** Select an existing login or create one.

To delete an application from the list, select the application, then click Delete.

The Description Column

Provides information about the application.

When you add an application that has a prebuilt script, the SecureLogin snap-in automatically enters a description for that application. You can edit this description.

When you manually enter an application name, the SecureLogin snap-in automatically sets the description to the name of the executable file or URL that you enter.

To edit a description:

- 1 Select the application, then click Edit.
- 2 Edit the Description field, then click OK.

Creating Logins

The Login Details window displays names (for example, GroupWise®) of applications that can use single sign-on.

A login is a collection of sensitive information, such as passwords. You define a login so that two different applications can share the same login information. For example, gmail.novell.com and grpwise.exe both use the GroupWise login.

To create a login name:

- 1 Click New, then enter a descriptive name in the New Login dialog box.
- 2 Click OK.

Synchronizing Applications and Logins

Adding Password Policies

A policy is a set of requirements or rules (for example, the number of characters required for a password). SecureLogin uses policies to enforce security during logins.

To add a password policy:

- 1 From the Password Policies tab, click New, type a name, then click OK.
For example, enter GroupWise Policy instead of GroupWise. Describe the application but don't use the name used on the Login Details page.

- 2 Add settings.

The settings include password length and case of letters.

Select the policy name, click Edit, enter settings, then click OK.

To set advanced settings, select the policy name, click Edit, then click Advanced. To assign Minimum and Maximum password character lengths, enter a number in the entry field.

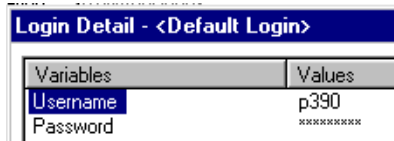
For details about settings, refer to the help system.

Working with Variables

SecureLogin stores your username and password in the form of a variable and its value. Your username and password are not included in the script. Instead, a variable is used in the script. The value of the variable is your username or password.

Logins consist of key-value pairs. The pair is a variable. You can use any name for the variable.

As the following figure illustrates, the Variable column usually just contains the password and username for a particular application.



Variables	Values
Username	p390
Password	*****

However, in some more complicated applications, there may be other variables too.

This example has two variables, Username and Password. The script for this platform has the following line:

```
type $Username
```

The variable `$Username` is written in the script. The value of `$Username` is `p390`. When the script runs, SecureLogin looks for the variable `$Username` in the user's login details. There it finds and reads the value `p390`. SecureLogin enters the value `p390` into the login panel. At run time, the value of the variable `$Username` (`p390`) is read. However, in the script we only see the variable `$Username`.

To enter a variable:

- 1 Click New, then name the variable (for example, Password)
- 2 Click OK.

After you enter a variable, you can't change that name.

To add or edit a value:

- 1 Select the variable, then click edit.
- 2 Type a value, then click OK.

For example, enter 1001 for the location of the password field on a login screen.

To display your password for this login, check the Display Passwords check box.

To delete a variable and accompanying value, select the variable, then click Delete.

For more information on variable substitution, see [Variables](#) in the *SecureLogin Script Commands* guide.

Creating a Script

Each application has a script. The script tells SecureLogin what to do concerning the application.

For example, a script for a Windows application specifies the executable filename, the controls, and information about dialog boxes.

A script for a Web application specifies the URL and fields to fill in. SecureLogin matches the URL name on the Applications page with the URL on the Web and then runs the script.

For more information, see [Chapter 6, "Administering Scripts," on page 93](#). For tips, see ["Troubleshooting Scripts for Web Sites" on page 152](#).

Using Policies to Tighten Security

Setting Policy Requirements

To tighten security, you can control the data (for example, the length of passwords) that users enter for logins.

- 1** In ConsoleOne, select a User or Organization object.
- 2** Right-click, click Properties, then click Password Policies.
- 3** Select a policy, then click Edit.
- 4** Enter values.

For details about each setting, refer to the help system.

The Advanced button enables you to set additional requirements.

Displaying Passwords

By default, the Login Details screen displays passwords as xxxxxxxx. You can view the actual password.

- 1** Select the application.
- 2** Check the Display Passwords check box.

Customizing SecureLogin

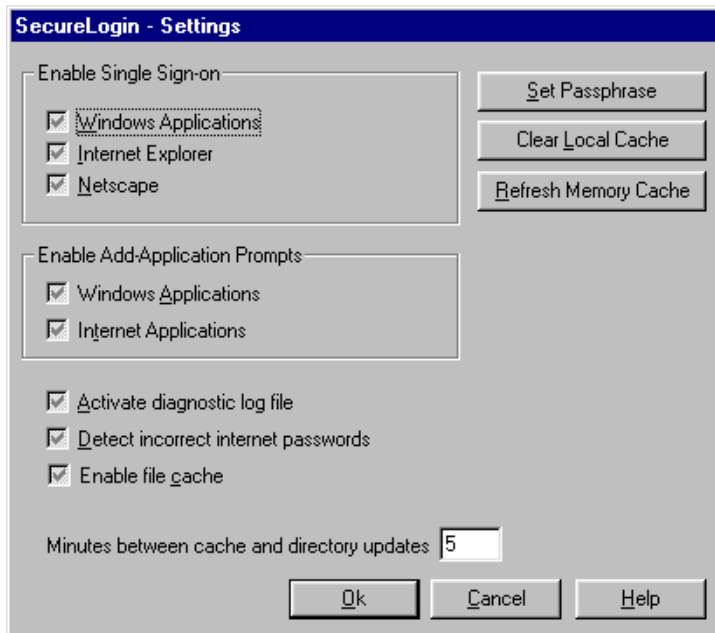
You and users can customize the operation of SecureLogin by using the following:

- ◆ The Change Settings option in the SecureLogin user tool.
- ◆ The User Preferences page on the SecureLogin snap-in to ConsoleOne.
- ◆ The Admin Preferences page on the SecureLogin snap-in.

The Admin Preferences page enables you to restrict users from accessing the Change Settings option as well as further customize SecureLogin behavior.

Using SecureLogin to Change User Settings

The following figure illustrates user settings that you can change by using the SecureLogin user tool:



A gray check box doesn't mean that the feature is disabled. Instead, a gray check box indicates that the setting is determined by configuration data on the network or (if configuration data exists) that the setting is at the default value See [“Understanding Default Settings” on page 79](#).

Enabling Single Sign-On

Windows Applications

To enable the Windows single sign-on features of SecureLogin, check the Windows Applications check box. To disable Windows single sign-on, uncheck the check box.

Default: On (checked)

Internet Explorer

To enable Microsoft* Internet Explorer features for SecureLogin, check the Internet Explorer check box. To disable Internet Explorer single sign-on, uncheck the check box.

Default: On (checked)

Netscape Login Active

To enable the Netscape* single sign-on features for SecureLogin, check the Netscape check box. To disable Netscape single sign-on, uncheck the check box.

Default: Enabled (checked)

Enabling Add-Application Prompts

Windows Applications

This Windows Applications setting controls whether the Windows single sign-on component automatically detects Windows login panels. To receive a prompt to run the wizard, check the check box.

Default: On (checked)

Internet Applications

The Internet Applications setting controls whether the Web single sign-on component automatically detects Web login panels. To receive a prompt to run the wizard, check the check box.

Default: On (checked)

Activating a Diagnostic Log File

To log the details of use to the hard drive, you can check the Activate Diagnostic Log File check box. However, because this preference is used for debugging and troubleshooting, do not enable this option unless Technical Services advises you to. Leave the check box unchecked.

Default: Off (unchecked)

Detecting Incorrect Internet Passwords

To enable SecureLogin to attempt to detect whether you have given it an incorrect Internet password, check the Detect Incorrect Internet Passwords check box. SecureLogin then prompts you to change the password.

Default: Off (unchecked)

Enabling File Cache

Username and passwords are normally stored in a directory on the server, but if the server is unavailable, or if you are using a notebook computer, the cache is used. The cache is password protected and encrypted.

To enable SecureLogin to use cache files, check the Enable File Caching for Office Use check box.

Default: On (checked)

Specifying Minutes between Cache and Directory Updates

The Updates rate controls the number of minutes that SecureLogin waits between synchronizing the information between the local cache and eDirectory.

Default: 5 (minutes)

Setting a Passphrase

The Set Passphrase option enables you to reset your selected passphrase and password combination.

- 1** Click Set Passphrase.
- 2** Enter the password to your passphrase, then click OK.
- 3** Enter a new passphrase question and answer.
- 4** Confirm the answer, then click OK.

Clearing the Local Cache

To clear the entries held in the local cache, click Clear Local Cache.

Refreshing Memory Cache

To force SecureLogin to immediately synchronize the data between NDS 7 (or later) or eDirectory and the local cache, click Refresh Memory Cache.

Toggling the Active Setting

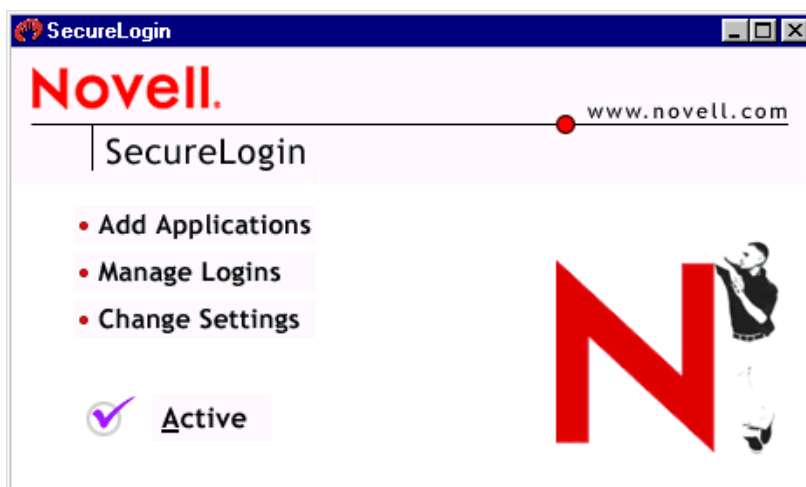
You can use the active button to override all the settings and disable all the SecureLogin modules.

Scenario: Writing a New Script. You have started writing a script for an application. Because you need to inspect some control IDs with Window Finder, you do not want the script to run at this time. You disable the Active setting. SecureLogin doesn't run while you fine-tune the script. After troubleshooting, you enable the Active button.

Scenario: Troubleshooting an Existing Script. You have a script already written for an application. You open the application to get more information from it, so that you can put the information in your script. Because the Active setting is enabled, the script for the application runs.

You don't want to wait for the script to run while you look for control IDs. Therefore, you close the application, turn off the Active setting, and open the application. You then get the information that you need, return to your script, enter the information into the script, enable the Active setting, and run the script.

The Active setting can also be useful for the help desk to use in troubleshooting, so that a technician can step through a login manually.



Using ConsoleOne to Change User Settings

You can use the SecureLogin snap-in to ConsoleOne to change all settings that are available in the SecureLogin user tool except the following:

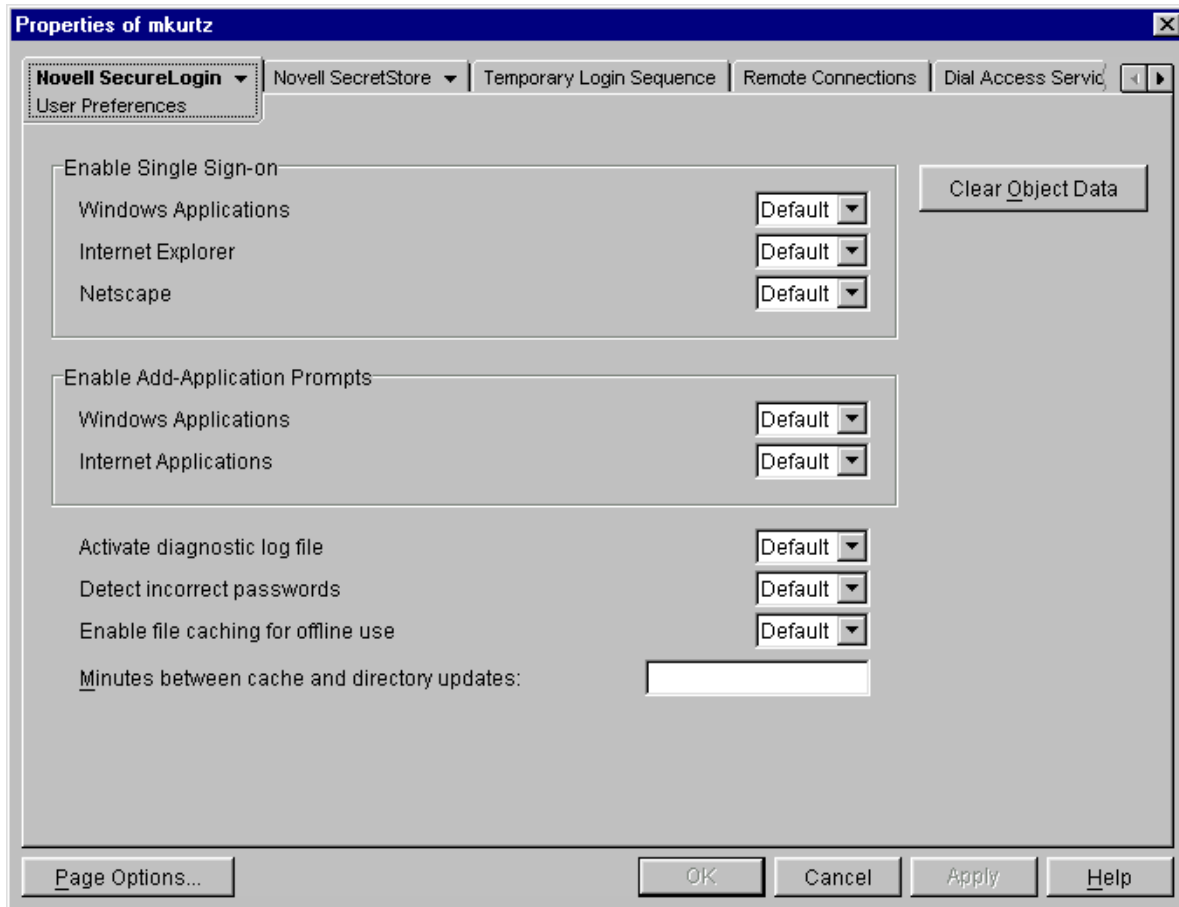
- ◆ Set Passphrase
- ◆ Clear Local Cache
- ◆ Refresh Memory Cache

However, the User Preferences page of the snap-in has one option not found in the user tool: Clear Object Data.

Clearing Object Data

If the SecureLogin eDirectory attribute needs to be refreshed, you can quickly clear the data and let it resync with the cache files. To remove the attribute, click Clear Object Data.

The following figure illustrates this option:



Understanding Default Settings

In ConsoleOne, the Novell SecureLogin tab has three options: Manage Logins, User Preferences, and Admin Preferences. For the two Preferences pages, the Default setting might be any of the following:

- ◆ Administrator-defined
- ◆ Product-defined
- ◆ User-defined

Administrator-Defined Settings

Whenever you (as administrator) change a setting, that setting becomes the Default setting. Other contexts inherit this setting. Even though you change a setting at a parent context, SecureLogin or ConsoleOne displays Default in subordinate contexts.

Scenario—Changing a Default Setting: At the Digital Airlines Company, you don't want anyone to view their passwords. In ConsoleOne, you select the Organization context, which is

digitalairlines. Then you select Properties, click Novell SecureLogin, then click Admin Preferences. From the dropdown list for Prevent Users from Viewing Passwords, you change the setting from Default to Yes.

At the RSDev context, you view the Prevent Users from Viewing Passwords setting, which displays Default. The default setting is actually Yes. You don't have to change Default to Yes (Prevent) because the setting is inherited from the parent context. Users can't view their passwords in the RSDev context.

Product-Defined Settings

If you (as administrator) don't change any settings for a context, the Default setting is the value defined by the application.

However, you can change settings on the User object.

Scenario—No Changes: At the Digital Airlines Company, you prefer that all users be able to view their passwords. You do not change any settings in ConsoleOne. At the digitalairlines context, the Default setting reflects the value that SecureLogin provides. Users can view their passwords.

At the RSDev context, you view the Prevent Users from Viewing Passwords setting, which displays Default. The default setting remains as set in SecureLogin. Because no administrator-defined setting exists, no setting is inherited from the parent context. Users can view their passwords.

User-Defined Settings

If you (as administrator) change a setting in ConsoleOne, that changed setting becomes the Default setting for users. Users can't change it. They can view the setting by using the Change Settings option in SecureLogin. However, they can neither view the Admin Preferences settings in ConsoleOne nor change the values that you have set administratively.

If you don't change a setting, the application's setting is the Default setting, unless the user changes it.

If the user changes a setting, that setting is user-defined.

In ConsoleOne, the Default setting on the User Preferences page could mean administrator-defined, product-defined, or user defined. What happens at the user level depends on what the administrator does or does not allow as well as what the product has defined as default values. To determine the value, you must go to the parent context and setting.

Scenario: In ConsoleOne, you select the digitalairlines context > Properties > User Preferences. To control Enable Single Sign-on functionality, you set the following:

Parameter	Setting
Windows Application	Yes
Internet Explorer	No
Netscape	Default

Rie is in the RSDev context. You select the User Preferences page for Rie and view settings, which display as follows:

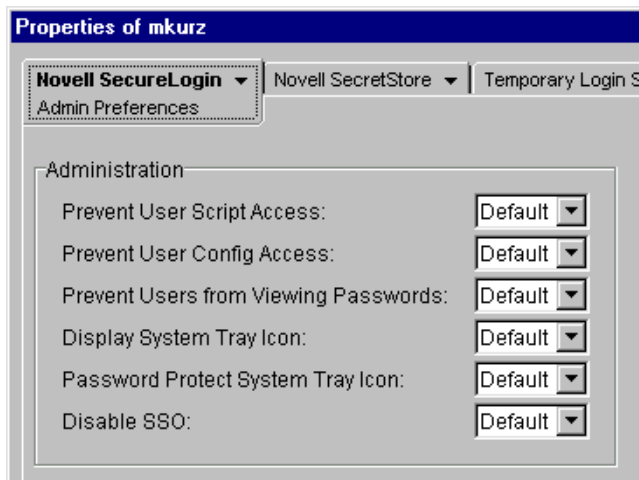
Parameter	Setting
Windows Application	Default
Internet Explorer	Default
Netscape	Default

Because you set Windows Application to Yes as administrator and at the Organization context Preferences, the setting is inherited. The Default setting for Rie is actually Yes. Because Internet Explorer is set to No at the Organization level, the Default setting for Rie is actually No. Because no setting was defined for Netscape at the Organization level, the Default setting is whatever value the product has determined.

Rie can change the setting for Netscape.

Using ConsoleOne to Set Administrative Options

As administrator, you can use the Admin Preferences option on the snap-in to ConsoleOne to control what users can do with SecureLogin at their workstations. The following figure illustrates this page:



To access this page:

- 1** In ConsoleOne, right-click an object (for an example, an OU or User object).
- 2** Select Properties.
- 3** At the Novell SecureLogin tab, select Admin Preferences.

Preventing Users from Accessing Scripts

The SecureLogin user tool has a Script page. This page enables users to view, create, or modify scripts for logging in to applications.

To prevent users from accessing scripts, select Yes at the drop-down menu for Prevent User Script Access. Users are then unable to use the New and Edit buttons on the Applications page.

Preventing Users from Changing Settings

The Change Settings option enables users to customize the SecureLogin environment at their workstations.

To prevent users from customizing the environment, select Yes at the drop-down menu for Prevent User Config Access.

Preventing Users from Viewing Their Passwords

The SecureLogin user tool enables users to view passwords that they use to log in to applications.

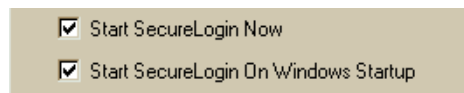
To prevent users from viewing passwords, select Yes at the drop-down menu for Prevent Users from Viewing Passwords.

To view a password:

- 1 At the SecureLogin main page, select Manage Logins.
- 2 At the Application tab, select an application name, then click Edit.
- 3 At the Logins tab, select an entry, then click Edit.
- 4 At the Login Details page, check the Display Passwords check box.

Displaying the System Tray Icon

During installation, you encountered a Post-Install screen that displayed the following options:



If you checked the Start SecureLogin on Windows Startup check box, SecureLogin places the SecureLogin icon on the system tray whenever you start the computer.

To prevent users from displaying and accessing the system tray icon, select No at the drop-down menu for Display System Tray Icon.

If you turn off the SecureLogin icon on the system tray and then refresh the data, the changes won't take effect until the workstation is restarted.

Password Protecting the System Tray Icon

You can require users to provide their passwords before they can access options on the system tray icon. Select Yes at the drop-down menu for Password Protect System Tray Icon.

Reading Corporate Scripts

By default, SecureLogin reads its information from the current user's context and then searches up the eDirectory tree. You can select where SecureLogin reads its corporate configuration and application information. At the Read Corporate Scripts From dialog box, browse to and select the desired context.

To prevent searching upward from the selected context, Select Yes at the drop-down menu for Stop Walking Here.

Using SecureLogin with Client Software

When you use SecureLogin with the following, additional functionality allows SecureLogin to leverage the user's eDirectory authentication information:

- ◆ The Novell Client™ for Windows NT* or the Novell Client for Windows 9x
- ◆ The eDirectory or eDirectory with SecretStore option

You can reference the *?sysuser* and *?syspassword* variables from within an application script, avoiding the need to store a copy of this data in the Directory. Additionally, when the user is authenticating to SecureLogin while disconnected from the Directory, the NDS or eDirectory password can be used in place of the user's passphrase answer.

If you use SecureLogin with the Novell Modular Authentication Services (NMAS™) client, using the NDS or eDirectory password method, no additional installation or configuration is required. By default, the NMAS client hides the NDS or eDirectory password field on the Novell Client login dialog box in favor of a subsequent password prompt.

To re-enable the Login dialog's password field:

- 1** Right-click the N icon on the system tray.
- 2** Select Novell Client Properties, then click Location Profiles.
- 3** Double-click Default in the Location Profiles window, then click Properties.
- 4** On the NMAS tab, check the Display Password Field by Default check box.
- 5** On the Credentials tab, check the Enable Password Field check box.
- 6** Close open dialog boxes by clicking OK.

To enable use of eDirectory login data without the NMAS client, you can install the SecureLogin client login extension (slina.dll) from the `\securelogin\terminalserver\citrix\client` directory.

- 1** Without replacing newer files, copy `slina.dll` and `unicows.dll` to the Windows SYSTEM directory (`c:\winnt\system32` or `c:\windows\system`).
- 2** Register the login extension by double-clicking the appropriately-named `.reg` file.
 - ◆ On Windows NT, Windows 2000, or Windows XP workstations, register the login extension by opening `Register NT LoginExt.reg`.
 - ◆ On Windows 98 workstations, open `Register 98 LoginExt.reg`.

Administering SecureLogin for Active Directory

The SecureLogin snap-in to Microsoft Management Console (MMC) enables you to administer SecureLogin for Active Directory*.

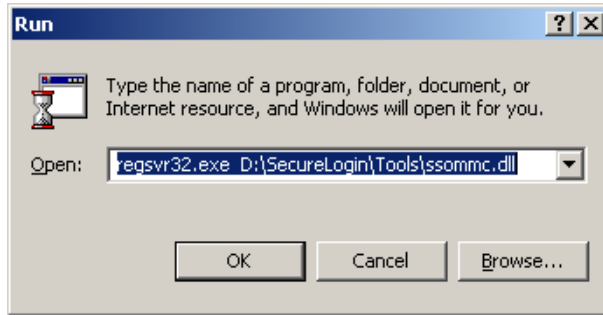
- 1** Make sure that SecureLogin client is installed on the FSMO server.
See [“Installing SecureLogin on an Active Directory Server” on page 27](#).

- 2** Register `ssommc.dll` with `regsvr32.exe`.

`SSOMMC.DLL` is in the `\securelogin\tools` directory on the CD.

To register `ssommc.dll`, use a batch file or an installation program.

The following figure illustrates selecting this file:



- 3** Copy ssommc.dll from the \securelogin\tools directory to Program Files\novell\securelogin.
- 4** Open the Microsoft Management Console with the Active Directory Users and Computers snap-in.

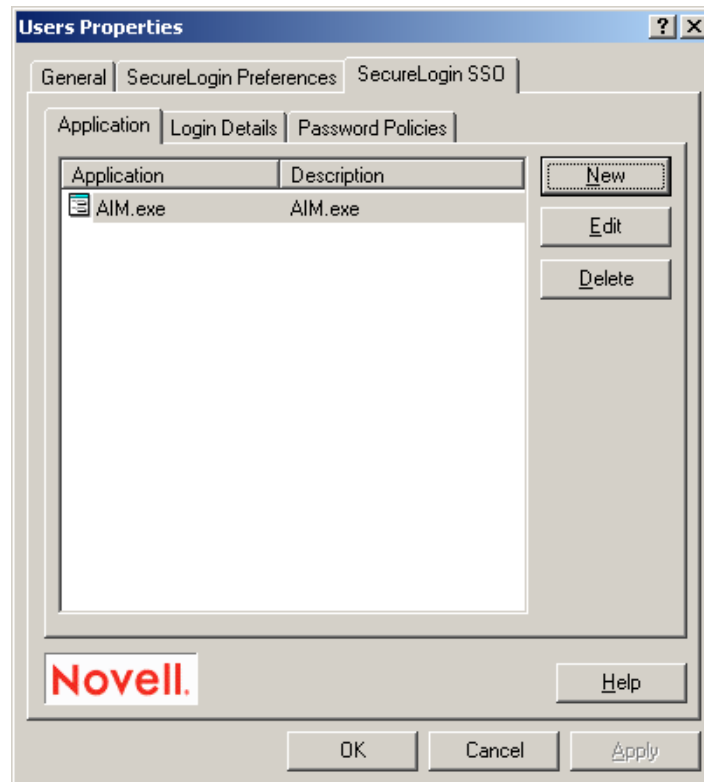
You saved the MMC as a filename. See [Step 7 on page 28](#).

- 5** Right-click a Container or User object, then click Properties.
- 6** Select a tab and edit settings.

When working with both of the following tabs, select the SecureLogin SSO tab first.

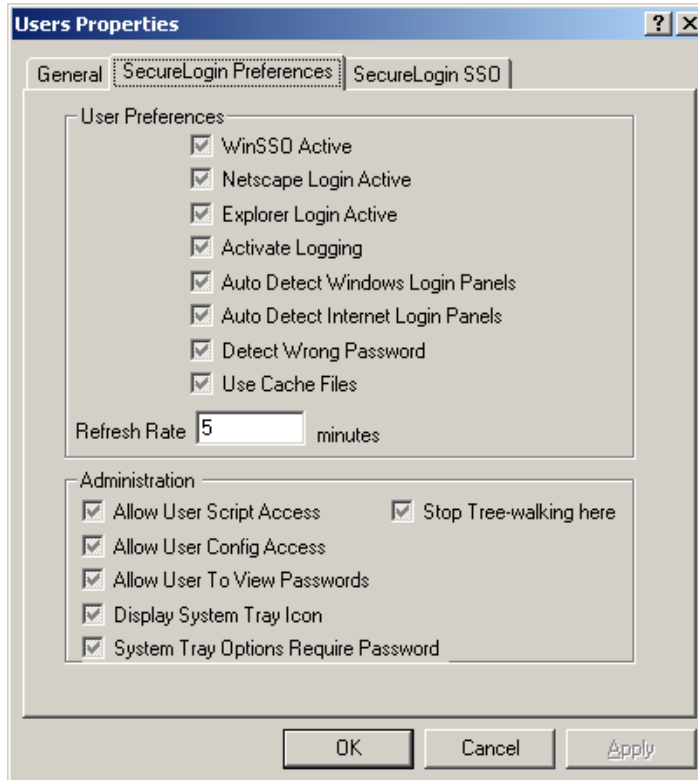
- ◆ SecureLogin SSO

The following figure illustrates the SecureLogin SSO tab:



- ◆ SecureLogin Preferences

The following figure illustrates the Preferences tab:



You administer the settings just as you do for ConsoleOne. See [“Customizing SecureLogin” on page 75](#).

Sharing Secrets

SecureLogin can share secrets between applications. See [“The Need for Shared Secrets” on page 49](#).

SecureLogin can also share secrets with other software solutions, such as Novell Portal Services (NPS) and iChain. For example, after you configure a Web site for SecureLogin, NPS can use the secrets in eDirectory to access that Web site.

In addition, when you change a password in either SecureLogin or NPS, the other software service recognizes and uses that changed password.

So that both SecureLogin and NPS can share a secret for an application, provide a common name for that application. Then refer to that common name when configuring the application for SecureLogin, iChain, or an NPS gadget.

For an example configuration of NPS and SecureLogin sharing secrets, see [Example Configuration: Sharing Secrets with Novell Products](#), in the *Novell SecretStore Administration Guide*.

Setting Up Multiple Logins for an Application

Scenario: Multiple Identities. Henri typically accesses aruser.exe as user Henri. However, Henri must occasionally access aruser.exe as user Admin. Therefore, Henri’s job responsibilities require

that he have two identities for the application aruser.exe. Henri sets up an identify for each role. SecureLogin automatically logs Henri in to aruser.exe according to the role that he selects.

To set up multiple logins for an application:

- 1 (Conditional) Create a login for an application.

If a login already exists for the application that you need multiple logins for, skip this step.

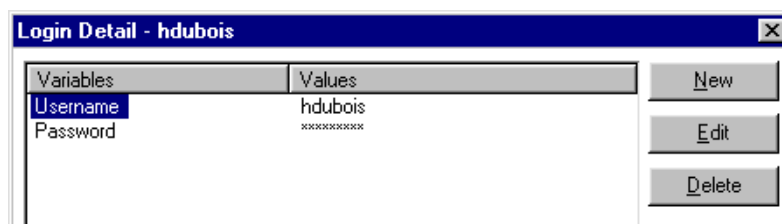
The application can be a Windows, Web, or other application. The application has a username and password.

- 2 Right-click the SecureLogin icon on the system tray, then click Add New Logon.

In the Available Applications pane, SecureLogin then displays all applications that SecureLogin has been configured to work with.

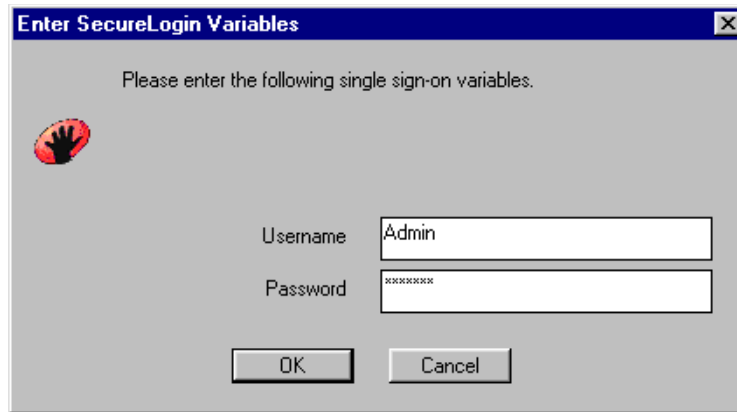


- 3 Click the application that you want to create an additional login for, then click Next.



- 4 Type a differentiating description for the new login that you are creating, then click Finish. For example, type Remedy Admin.

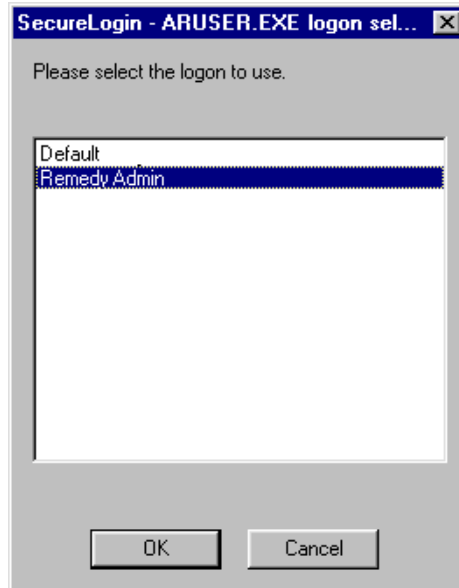
- 5 Type variables for the new login, then click OK.



Type the username and password for the role that you are creating the new login for.

- 6 For subsequent logins to the application, select the login that you need.

As the following figure illustrates, a list displays each login that you have created. The Default option is for the first login that was created.



Each login has a separate set of variables. The script for the login could use two variables (for example, a username and password) or more than two. Also, the variables might be named something other than Username and Password.

Changing the Startup Order of Applications

If a password-protected application starts before SecureLogin is initialized, SecureLogin is unable to process the login request for that application. To solve this problem, change the startup order of the applications. Use one of the following options, according to how your application has been configured to autostart:

- ♦ [“Using Startup Scripts to Start Applications” on page 88](#)
- ♦ [“Using Novell Application Launcher to Start Applications” on page 89](#)

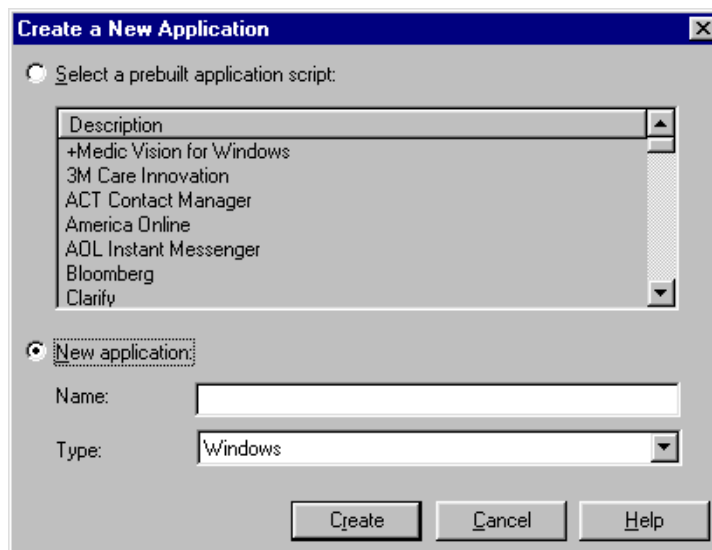
Using Startup Scripts to Start Applications

Some installation programs place application startup entries in the following locations:

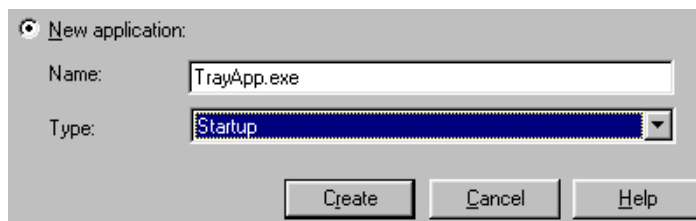
- ◆ The Windows Startup program group (for example, Start Menu\Programs\Startup)
- ◆ The Windows Run registry keys (for example, HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run)

If these applications require passwords, you can remove the startup shortcuts or registry keys and use a startup script to launch the applications. SecureLogin can then receive and process login requests for these applications.

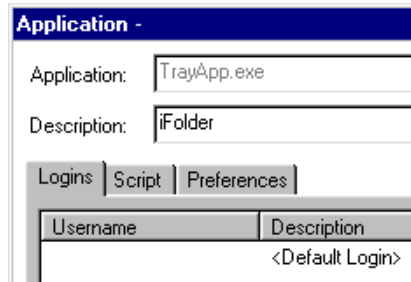
- 1** Right-click the SecureLogin icon on the system tray, then click Manage Logins.
Select Startup as the type.
- 2** At the Applications tab, click New.
- 3** Click New Application.



- 4** In the Name box, type the name of the executable (for example, TrayApp.exe)
- 5** In the Type box, select Startup, then click Create.

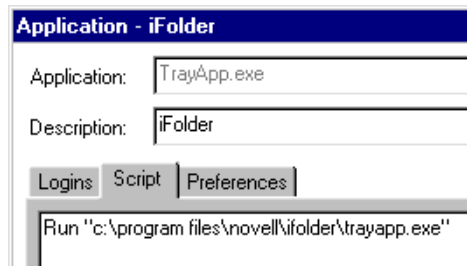


- 6** In the Description box, type the name of the application (for example, iFolder).



- 7 Type a script, then click OK.

Click the Script tab, then type the script. As the following figure illustrates, you should include the Run command, the path, and filename of the executable.



If the path includes a space (as in program files), enclose the path and filename in quotation marks. Otherwise, omit the quotation marks.

- 8 (Conditional) If you haven't already created a login for the application, create one now. See [“Creating Logins” on page 73](#).
- 9 Remove the application's shortcut from the StartUp program group or from the Run registry key.

Because the startup command is now in a SecureLogin Startup script, the application will launch after SecureLogin is loaded and able to handle login requests.

Using Novell Application Launcher to Start Applications

Use the Icon Order and Wait on Force Run options in Novell Application Launcher™ (NAL). These options enable you to use NAL to do the following:

- ◆ Launch Novell SecureLogin.
- ◆ Launch (with a lower order) the other applications that you want to single sign-on to at startup.

One possible drawback with this option is that some users might not want to start an application that NAL launches.

Using Login Watcher

Login Watcher helps you enable applications to work with SecureLogin.

To use Login Watcher:

- 1 Close SecureLogin.

SecureLogin might prompt you to close some applications before it can stop running.

- 2 Run Login Watcher (securelogin\tools\loginwatch.exe).
- 3 Enter the name of the application you want to watch (for example, aruser.exe).



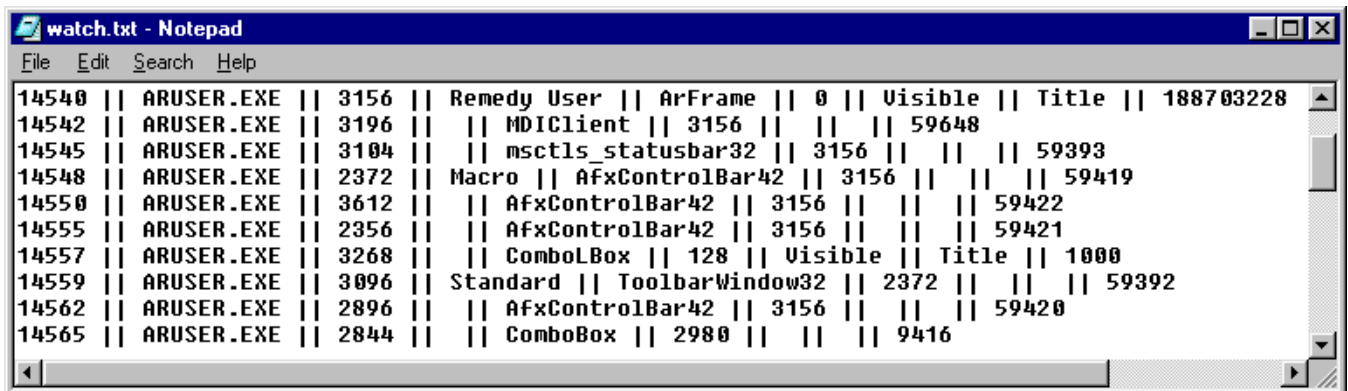
Include the filename extension.

- 4 Click Start.

The following dialog box appears in the corner of the screen.



- 5 Run the application (for example, Remedy Aruser) that you want Login Watcher to watch. Login Watcher logs all details related to the dialog boxes that the program displays.
- 6 Click Stop on the Login Watcher dialog box.
- 7 Click View Log.



The log is saved in c:\watch.txt.

Login Watcher records the information in the following format: time || ModuleName || window handle || window text || class name || parent || Visible flag, Title Flag || Control ID.

Time—The milliseconds elapsed since the recorder was started (for example, 14540).

Module Name—The name of the executable that created the window (for example, ARUSER.EXE).

Window Handle—The unique identifier of that instance of the window (for example, 3156).

Window Text—The text of the window (for example, Remedy User).

- ◆ For edit boxes, the text is the contents.
- ◆ For buttons, the text is the label.
- ◆ For windows with titles, the text is the title.

Class Name—The name of the window class (for example, ARFrame).

Parent—The window handle of the parent window.

The window handle allows SecureLogin to cross-reference through the list and find the parent.

Visible Flag—Set on top level windows that have the style Visible set.

Title Flag—Set on top level windows that have the style Title set.

Control ID—The unique identifier of that control in the program, used for typing etc.

Login Watcher appends data to the existing watch.txt file.

6

Administering Scripts

This section contains information on the following:

- ◆ “Structuring and Executing Scripts” on page 93
- ◆ “Types of Scripts” on page 94
- ◆ “Managing Scripts” on page 94
- ◆ “Finding Dialog Control IDs” on page 95
- ◆ “Excluding or Including Specific Applications” on page 97
- ◆ “Scripts for Predefined Applications” on page 99
- ◆ “Creating Corporate Scripts” on page 100

Structuring and Executing Scripts

A script is a simple piece of text that is stored by the SecureLogin script broker. Scripts store the login name, password, and any other information in fields required for authentication. Scripts are stored in the local database and in eDirectory.

Each script has a name, called the application name, which uniquely identifies it within a particular single sign-on database. In addition, each script has a type, known as the application type (prebuilt, Windows, or Web). The application type specifies the type of application the script refers to and which of the SecureLogin components executes it.

SecureLogin scripts execute sequentially from the first line. There are no flow control mechanisms as such. There are, however, instances where a component might choose not to execute certain statements, as in the Dialog / EndDialog statement.

Each line in the script consists of one or more arguments. Arguments are separated by white space (spaces and tabs), unless they are enclosed in quotation marks. For example, the following line contains three arguments: A simple “command to get started”.

1. A
2. simple
3. “command to get started”

After a script has been broken into arguments, the quotation marks are removed. If you need to specify an actual quotation mark in a script, precede it with a backslash (for example, \").

The first argument on a line is the command. It specifies the action the line takes. The rest of the arguments on the line, if any, are passed to that command. Different commands take varying numbers of arguments. For a list of commands and their arguments, see [SecureLogin Commands](#) in *Script Commands*.

A line that begins with a # character is treated as a comment and is ignored in the script language. The following example illustrates the use of the # character:

```
Window "login"  
Delay 30  
#SecureLogin ignores this line and the next three lines  
#while executing the script.  
#The Delay is used to wait for the window to be created #correctly.  
Type "$Username"
```

Scripts are interpreted as SecureLogin components to perform the sign-in process. This functionality ensures that any variables that are substituted are current.

Types of Scripts

The following table lists types of scripts that SecureLogin supports.

Type of Script	Description
Windows* Applications	For Windows-based applications.
Terminal Launcher	For applications that require access via an emulator.
Internet Page	For Web-based applications.
Password Policy	For a password policy associated with one or many applications.
SecureLogin Startup	For the execution of a script during the startup of SecureLogin.
Corporate Scripts	Corporate scripts can be any of the other five scripts listed. Corporate scripts are indicated on the client by the letter C next to the icon for that script.

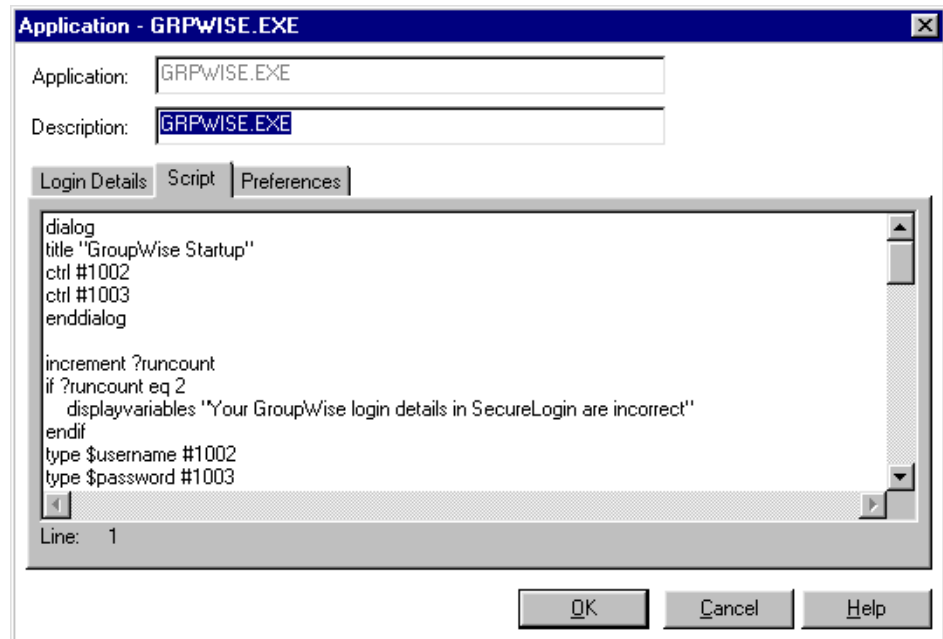
Managing Scripts

Each single-sign-on-enabled application has a script. A basic script tells SecureLogin how to log in to the application. More involved scripts can be created that allow you to perform other password management tasks, such as detecting expired passwords and generating new passwords.

You can manage scripts for applications by using ConsoleOne™ or the SecureLogin user tool. The following process is for ConsoleOne.

- 1 Select an object (for example, an OU or User object), then click Properties.
- 2 Select Novell® SecureLogin > Manage Logins, then click Applications.
- 3 Select an application, click Edit, then click Script.

The following figure illustrates the Script tab and the script for GroupWise®:



4 Make changes.

For commands used in scripts, along with sample scripts for those commands, see [SecureLogin Commands](#) in the *SecureLogin Script Commands* guide.

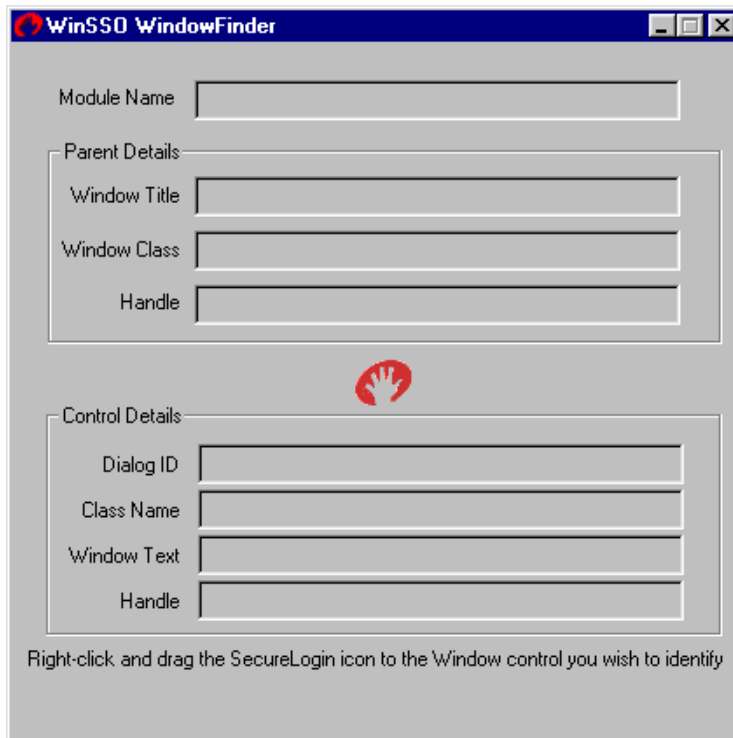
For troubleshooting tips, see [“Troubleshooting Scripts for Web Sites”](#) on page 152.

For a scenario to enable single sign-on authentication to MyRealBox, see [Using Novell SecureLogin to Enable Web Applications for Single Sign-On](#) (<http://developer.novell.com/research/appnotes/2002/may/02/apv.htm#1228584>) in the May 2002 issue of *AppNotes*

Finding Dialog Control IDs

A control ID is a number that uniquely identifies a field, such as a button, within a window. Many script commands related to logging into Windows applications require a dialog control ID.

To help you determine these control IDs, SecureLogin includes a tool called the Window Finder.



To inspect a control:

- 1 Click Start > Programs > Novell SecureLogin > Window Finder.
- 2 Right-click the SecureLogin icon and drag it over the control of interest.

The Window Finder tool displays the details of the control.

If an application page hides the Window Finder, click the WinSSO Window Finder icon on the system tray.

Module Name: The name of the executable that created the window.

Use this name for the application name of the Windows single sign-on script.

Window Title: The title of the window that contains the control.

You can use this title in a window or title statement.

Window Class: A field for information only.

Each window has a class associated with it.

Dialog ID: A unique identifier.

Each control has a unique identifier called the control ID. Use this number as the target for Type, Click, Ctrl, and SetPlat statements. For information on each of these commands, see [SecureLogin Commands](#) in the *Script Commands* guide.

Class Name: A name that determines the type of the control.

For single sign-on to work correctly, the SecureLogin Windows component must be able to read and write text to the specified control. The class name determines the type of the control and whether reading and writing is possible. Supported classes include edit, combobox, and static.

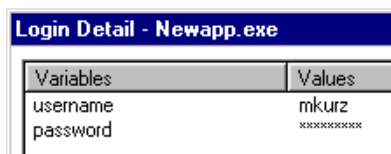
Window Text: A field that displays the text contained within the control.

This information can be useful in troubleshooting and for writing the regular expression required by the Setplat command.

Working with Variables

SecureLogin stores your username and password in the form of a variable and its value. Your username and password are not included in the script. Instead, a variable is used in the script. The value of the variable is your username or password.

Logins consist of key-value pairs. The pair is a variable. You can use any name for the variable. As the following figure illustrates, the Variable column usually just contains the password and username for a particular application. However, in some more complicated applications, there may be other variables.



Variables	Values
username	mkurz
password	xxxxxxxx

This example has two variables: password and username. The script for this platform has the following line:

```
type $username
```

The variable \$username is written in the script. The value of \$username in this example is mkurz. When the script runs, SecureLogin looks for the variable \$username in the user's login details. There it finds and reads the value mkurz. SecureLogin enters the value mkurz into the login panel.

At run time, the value of the variable \$username (mkurz) is read. However, in the script we only see the variable \$username.

Excluding or Including Specific Applications

SecureLogin's Exclude mode enables you to exclude Windows applications from detection. SecureLogin's Include mode enables you to limit detection of Windows applications to specified applications. By default, SecureLogin is set to Exclude mode for Windows single sign-on functionality.

NOTE: You can't simultaneously use Exclude and Include modes.

Excluding Applications from Detection

SecureLogin has a built-in list of Windows applications that are excluded from detection:

```
COMBRO~1.EXE  
combroker.exe  
loginw32.exe  
loginw95.exe  
MSDEV.exe  
nlnotes.exe  
notes.exe  
nswebsso.exe
```

Nwadm32.exe
Nwadm95.exe
Nwadmnt.exe
NWTray.exe
proto.exe
scrnlock.scr
setup.exe
tlaunch.exe

To add applications to this default list:

- 1 Create an exclude.ini file in the SecureLogin installation directory.

For example, create c:\Program Files\novell\securelogin\exclude.ini.

- 2 List the executable names for applications that you want to add.

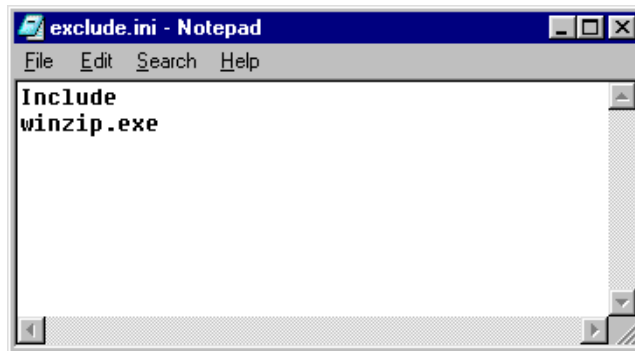
Add one executable name per line. For example, to prevent SecureLogin from detecting weblog.exe, enter the following line in the exclude.ini file:

```
weblog.exe
```

Including Applications for Detection

By using an exclude.ini file, you can specify and limit the programs that SecureLogin detects.

- 1 Create an exclude.ini file.
- 2 Place the Include command at the top of the file.



- 3 List the applications that you want SecureLogin to detect.

When using the Include command in the Exclude.ini file, you must list every application you want SecureLogin to detect. Using the Include command in the exclude.ini file does the following:

- ◆ Increases performance
- ◆ Stops the auto detection of new applications
- ◆ Stops the auto detection of SSO to any applications not listed in the file
- ◆ Increases security

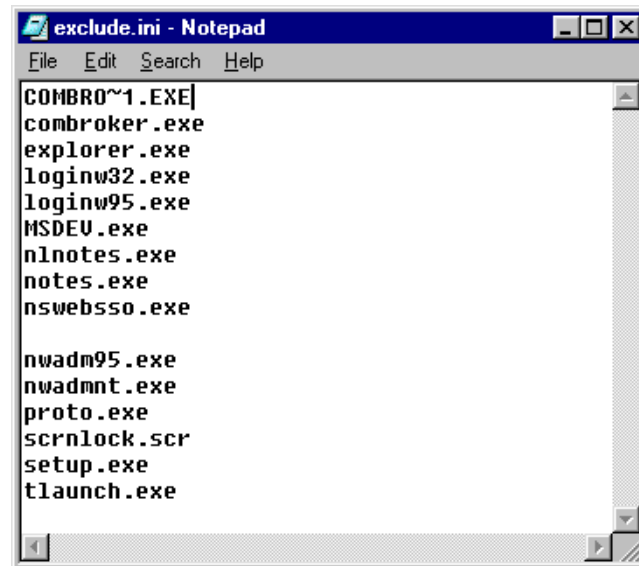
- 4 Save exclude.ini on the workstation in the SecureLogin installation directory.

Typically, the directory is c:\Program Files\Novell\SecureLogin).

If an application has been hardcoded to be ignored, you can have SecureLogin detect it.

- 1** Create an exclude.ini file.
- 2** Place the Nodefault command at the top.
This command causes SecureLogin to ignore the hardcoded list.
- 3** Rebuild the hardcoded list in the exclude.ini file, including all the applications in the list except the application that you want SecureLogin to detect.
- 4** Save exclude.ini on the workstation in the SecureLogin installation directory (typically c:\Program Files\Novell\SecureLogin).

For example, the following image removes NWadmin32.exe from the list of applications, allowing SecureLogin to detect Nwadmin32.



Scripts for Predefined Applications

SecureLogin provides native script support for many popular applications so that you don't have to configure them manually.

Application	Application	Application
+Medic Vision for Windows	MeetingMaker*	plusw33.exe
3M Care Innovation	Microsoft* Front Page	QuickBooks Pro*
ACT Contact Manager	Microsoft Internet Gaming Zone (lobby.exe)	Quicken
America Online*	Microsoft Internet Gaming Zone (zone.exe)	Remedy* ARUSER
AOL Instant Messenger*	Microsoft Money 98/99	Remedy Notifier
Bloomberg	Microsoft Networking Client	RiskMaster v3.6
Centra Symposium*	MMIS	SAP /R3 Login

Application	Application	Application
Clarify	MMIS (NTVDM)	Siebel* Customer Tracking
Corporate Time	Mobile UP v4.5	Soft Front
Entrust* Client	MS SQL	SoftMED Application Suite
Entrust Server	MSN Messenger	Solaris* Login
Eudora* Email	MYOB Premier	STARS
GoldMine*	Netscape*	Sunrise Clinical Manager
GoldMine 5.5	Novell BorderManager VPN Client	Visual SourceSafe Login
ICQ	Novell GroupWise Client	Windows 9x Dialup Networking
Informix Connect for Win32	Novell GroupWise Notify Client	Windows 9.x Login
Internet Explorer	Oracle* Generic Login	Windows NT* Logon
Lotus Notes*	Oracle Financials	Yahoo! Messenger
Lotus Organizer* 4 + 5	PCAnywhere 8.0	Zainetbar
Meditech Remote Workstation	PeopleSoft*	

Creating Corporate Scripts

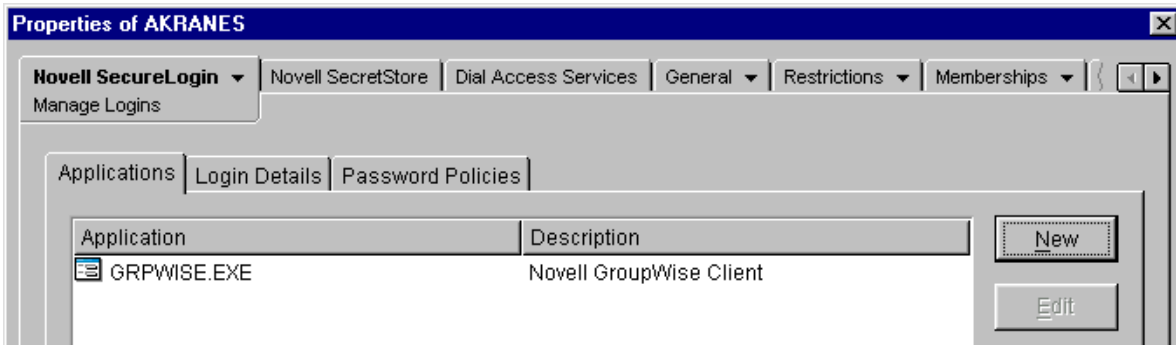
Corporate scripts are normal scripts that are assigned to a Container object instead of to a User object.

Because they are automatically rolled out to all User objects held in the Container object, corporate Scripts simplify implementing and administering SecureLogin single sign-on. By using this method, you don't have to configure applications for each individual user in your organization. All users read and use the same scripts.

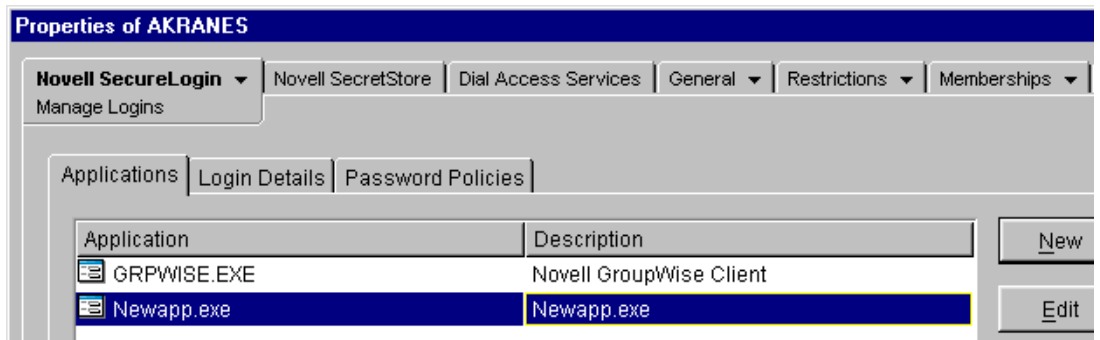
Windows Application, Web, Startup, and Terminal Launcher scripts can all be implemented as corporate scripts.

In a Microsoft ADS environment, use the Microsoft Management Console (MMC) to create corporate scripts. In an NDS or eDirectory environment, use ConsoleOne to create corporate scripts. This section describes how to use ConsoleOne.

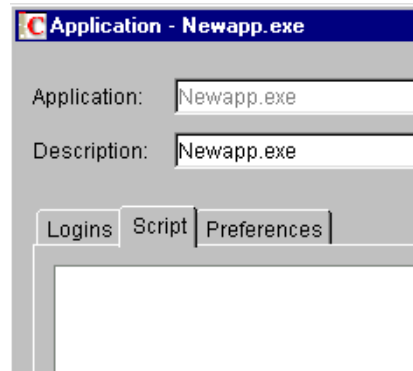
- 1** Log in as Admin.
- 2** In ConsoleOne, navigate to the Container object where you want to create the corporate script.
- 3** Right-click the Container object, then click Properties.
- 4** Select the Novell SecureLogin tab, then click New.



- 5 Select New Application, enter the executable name of the program that you want to create a script for, select the script Type, then click OK.
- 6 Double-click the new application in the application list.



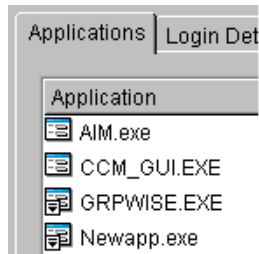
- 7 Select Script, then type the script into the text box.



For assistance with scripting, see [Script Commands](#).

- 8 Click OK > Apply > Close.
- 9 Browse the NDS or eDirectory tree and select a User object that is within the Container object that you just created the corporate script for.
- 10 Right-click the User object, then click Properties > Novell SecureLogin.

The User object has inherited the corporate script. Corporate scripts are distinguished from local scripts in the list by a small black triangle in the lower left-hand corner of the icon that sits beside the application name.



Exempting a User Object from a Corporate Script

Local scripts take precedence over corporate scripts. Occasionally, you might want a particular user to use a different script to the corporate script for a particular application. To do this, create a local script for the application at the User object level.

If you have a corporate script for an application, and you have a user that should not have that application single sign-on enabled, create a blank local script for the application at the User object level.

You can also use this procedure to exempt a Container object from corporate scripts inherited from Container objects that are higher in the directory tree.

7

Installing Terminal Servers

This section contains information about the following:

- ◆ “Integrating Microsoft Terminal Server and Citrix” on page 103
- ◆ “GINA Credential Pass-Through” on page 104
- ◆ “Integrating Citrix Components” on page 105
- ◆ “Virtual Channel” on page 107
- ◆ “Requirements” on page 108
- ◆ “Setting Up the Server” on page 109
- ◆ “Setting Up Workstations” on page 110
- ◆ “Virtual Channel Driver” on page 112
- ◆ “Registry Settings” on page 113
- ◆ “Debugging Options” on page 114
- ◆ “Integrating with Citrix-Published Applications” on page 114

Integrating Microsoft Terminal Server and Citrix

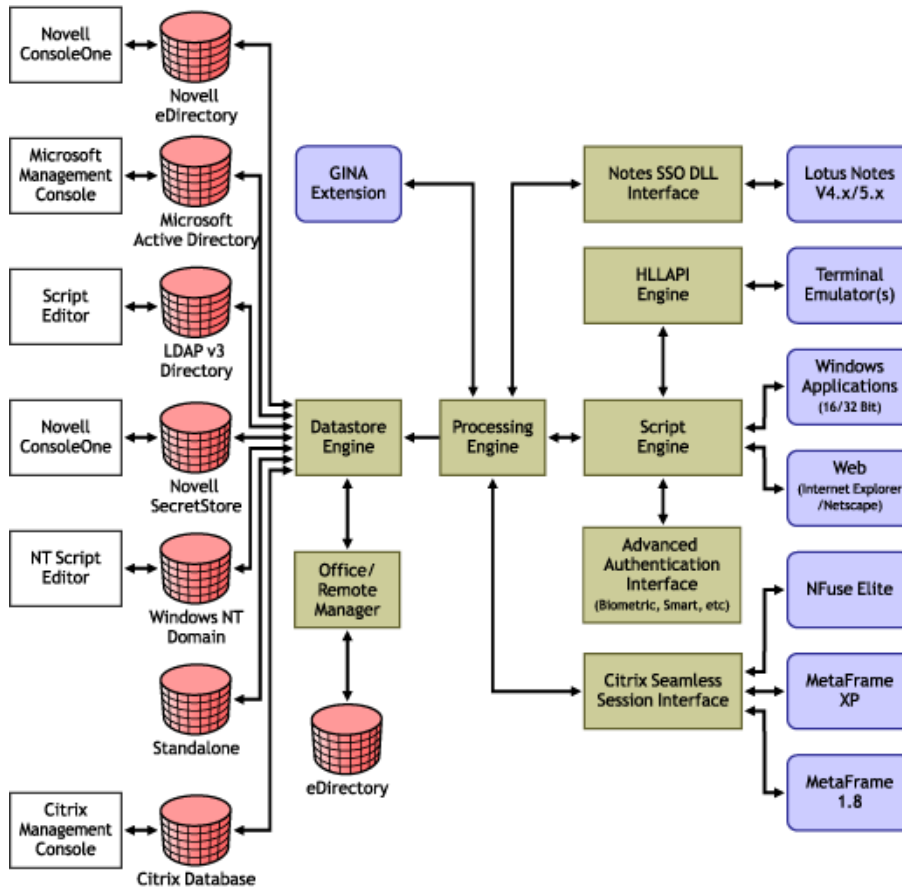
SecureLogin can simplify authentication to numerous configurations of Microsoft Terminal Server and Citrix MetaFrame*. Integration of SecureLogin Terminal Server and Citrix consists of the following components. Not all may be required, depending on your implementation.

- ◆ Client login extension (slina.dll) applied to a workstation with the Novell® Client™ (without the Novell NMAS™ client), which provides a link between the following:
 - ◆ The Novell GINA (graphical identification and authentication) or the Windows 9x login panel
 - ◆ The GINA running on the terminal server
- ◆ NMAS client integration library (slnmas.dll) applied to a workstation with the Novell Client and Novell NMAS client, provides a link between the client login and the GINA running on the terminal server

NOTE: Client slina.dll and slnmas.dll also provide support for offline authentication to SecureLogin by using your NDS or eDirectory username and password.
- ◆ GINA stub (sl_tscgina.dll) applied to a workstation without the Novell Client, which provides a link between the Microsoft GINA and the GINA running on the terminal server
- ◆ Server login extension (slina.dll) applied to a terminal server with the Novell Client, which provides the server-side link to the client GINA

- ◆ Server GINA replacement (sl_tsgina.dll) applied to a terminal server without the Novell client, which provides the server-side link to the client GINA stub
- ◆ SecureLogin Virtual Channel Driver (vdsllsso.dll or tsslso.dll), which provides the conduit for secure communications between the client and server extensions
- ◆ Published Application integration (ProLauncher.exe) applied to a Citrix server, which provides proper initialization and termination of the SecureLogin components (combroker.exe and proto.exe) running on the server

The following diagram illustrates the SecureLogin architecture:



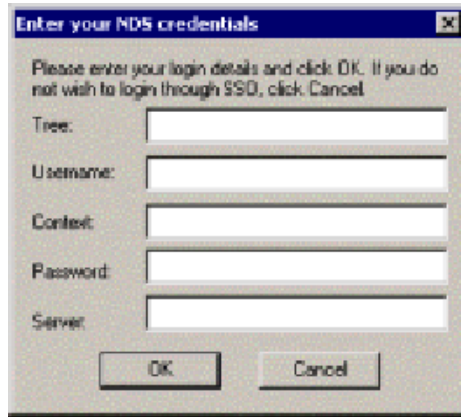
GINA Credential Pass-Through

With the SecureLogin Citrix components installed, SecureLogin provides a seamless pass-through of GINA credentials from the client to the server.

The GINA credential pass-through operates anytime that the terminal server presents a GINA login panel. If the credentials that the user used to log in to the client match the credentials of the terminal server, the credentials are automatically passed for the user.

If the credentials don't match, SecureLogin captures the error and presents a new login panel for the user to complete. SecureLogin detects which GINA is running on the Citrix server and requests the appropriate information.

For example, if SecureLogin detects that the terminal server has the Novell Client installed, SecureLogin presents the following dialog box:



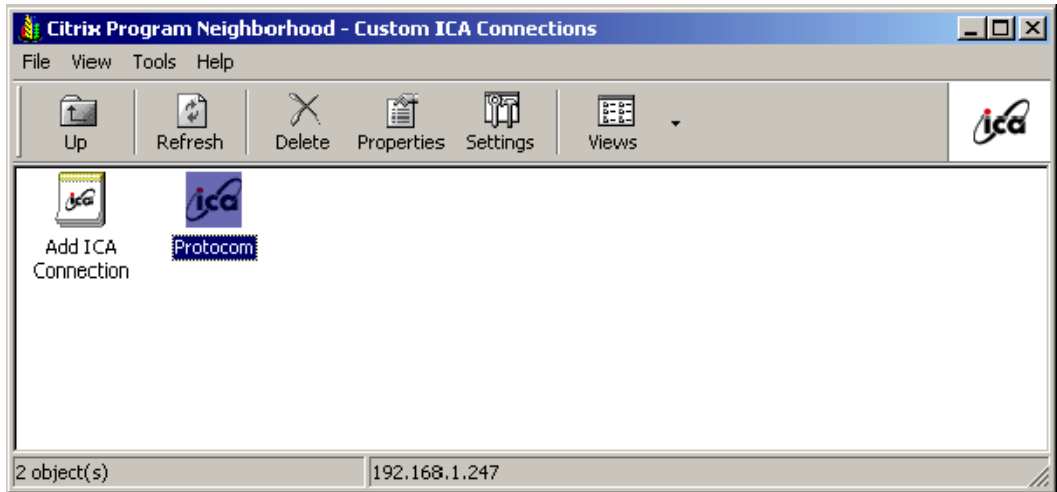
After the user completes the dialog box, SecureLogin saves the information as a hidden application (platform) within the SecureLogin datastore directory (and local cache if applicable). The next time the user accesses the terminal server, the credentials are retrieved from the hidden application and seamlessly passed to the terminal server.

Integrating Citrix Components

Citrix provides several ways to access a Citrix server or published application. How you access the server determines how SecureLogin handles the authentication to the server. Although different methods are used depending on how you access the server, all forms of authentication can be managed with SecureLogin.

Windows GINA Authentication

When the Citrix server requests a Windows GINA authentication, the Citrix Seamless Session Interface provides the credentials by using the hidden application. An example of this type of authentication occurs when you connect to a Citrix server through Program Neighborhood's Custom ICA Connection interface:



Another example of this type of authentication occurs when you export a published application to an .ICA file and distribute it to your workstations. This type of authentication is enabled by installing the GINA components. The authentication is not disabled even if SecureLogin is not currently active.

Program Neighborhood

When a user accesses a Citrix Farm using Program Neighborhood, Program Neighborhood uses WFCRUN32.EXE and presents a Program Neighborhood authentication dialog box:



Program Neighborhood then collects the credentials and sends them to a Citrix server in the farm.

The Citrix Seamless Session Interface does not handle this authentication request. However, the wfcrun32.exe file can be handled by a script just like any other Windows application that is requesting authentication. The SecureLogin Wizard automatically creates a script that enables SSO for Program Neighborhood. You should modify this script to allow for error handling, such as a bad username, domain, or password.

Using Desktop Shortcuts to Published Applications

If the Citrix Farm is configured to push out shortcuts to the user's desktops, the shortcut actually calls an executable pn.exe (for example, "C:\Program Files\Citrix\ICA Client\pn.exe"). Like wfcrun32.exe, this authentication is handled just like any other Windows application by using a script for pn.exe.

The SecureLogin Wizard automatically creates a script that enables SSO for pn.exe. Be sure to include error handling in case the user passes the wrong information into the dialog box.

Handling Password Changes

The Citrix Seamless Session Interface currently does not detect if users change their domain or NDS[®] or eDirectory[™] password through a Citrix connection. If a user changes this password through a Citrix connection, the interface detects the failed seamless authentication the next time that the user connects to the Citrix server. The interface then once again prompts the user for credentials.

When the user enters the correct (new) password, the interface saves that new password in place of the previous password in the hidden application within the applicable datastore (and local file cache if applicable).

Virtual Channel

A virtual channel is a session-oriented and bidirectional error-free transmission connection that can be used by application layer code to exchange custom data packets between a terminal server and a terminal client.

SecureLogin employs this technology to allow users to single sign-on to various Published Application or Remote Desktop logins.

Virtual Channel Components

SecureLogin Terminal Server single sign-on has three major components:

Component	Description
Client login extension	Collects users' login credentials for single sign-on
Virtual channel driver (VCD)	The heart of SecureLogin Terminal Server single sign-on. Liaises between the server login extension and single sign-on to perform all terminal session single sign-on processes
Server login extension	Requests users' login credentials from the VCD and initiates the login process. After authentication, the login extension returns credentials to the VCD to update single sign-on.

The three components use the following process:

1. A user enters a username and password, a domain (optional), an NDS or eDirectory context, and an NDS or eDirectory tree. This information is encrypted and stored in the registry.
2. SSO Combroker consumes the registry information and destroys the data in the registry. Login credentials are saved under a generic and hidden platform name.

3. When the user starts the Citrix ICA client or a published application through an ICA file, the SecureLogin virtual channel driver is loaded. This driver receives the domain or preferred tree name of the server. To retrieve the username, password, domain, NDS or eDirectory context, and tree, the driver then reads the platform name Combroker.

If the platform does not exist, the VCD reverts to the generic platform name.

If the generic platform name does not match the requested platform (tree or domain), the VCD displays a dialog box to prompt the user to enter NDS or eDirectory or NT credentials. The expected credentials depend on whether the request is coming from a server with a Novell Client or from an NT/2000 server. The collected credentials are then sent to the server for verification.

When the user enters and accepts the credential dialog box, a hidden application is created for the next authentication request.

If the user chooses to abort entering credentials, the server login box appears as usual.

NOTE: SecureLogin does not currently handle the actual password change process. Therefore, SecureLogin does not send back the new password when changed on the Citrix server. However, when the password stored in Combroker is invalid due to a recent password change done on the Citrix Server, the user will be prompted to enter login credentials again. After the new password is verified, it will then be sent back to VCD to update the Combroker.

4. After a successful authentication, the server login extension always sends the user's login credentials back to the workstation. If an application does not exist, this procedure creates a new application in Combroker. If the password has recently been changed and the application already exists, this procedure updates the new password to Combroker.

Auto-Detecting the Client Protocol

The server detects whether the ICA protocol is present or not. If the ICA protocol is present, the server loads the ICA protocol. If the client is trying to establish a session by using the RDP protocol, the server loads the RDP protocol and the session begins. After the server is installed, it automatically responds to the RDP or ICA protocol.

By default, the Auto Detection feature is on.

Windows NT 4.0 Terminal Server Edition (RDP 4.0) does not support the virtual channel operation. If the client tries to establish a session by using the RDP protocol, Windows NT 4.0 Terminal Server Edition won't respond to the client.

Requirements

Server

- ◆ Windows NT 4.0 Terminal Server Edition or Windows 2000 Server family with Terminal Service enabled

NOTE: Only the Windows 2000 server family operating systems support Virtual Channel. If you want Virtual Channel support on Windows NT 4.0 Terminal Server Edition, you must install an appropriate Citrix server.

- ◆ One of the following Citrix servers installed (optional):
 - ◆ MetaFrame 1.8 for Windows 2000
 - ◆ MetaFrame 1.8 for Windows NT 4, Terminal Server Edition

- ◆ MetaFrame 2.0
- ◆ Citrix XP
- ◆ Novell Client version 4.7 or later (optional)

NOTE: Only Windows 2000 Server family operating systems support Virtual Channel. If you want virtual channel support on Windows NT 4.0 Terminal Server Edition, you must install an appropriate Citrix server.

Workstation

- ◆ Novell Client version 4.7 or later

If you use the Novell Client, use version 4.7 or later. The Novell Client is not required unless you are using Windows 95.
- ◆ SecureLogin version 3.0.1 or later
- ◆ One of the following:
 - ◆ Win32 ICA Client Version 6.00.905 or later
 - ◆ Terminal Server Client that supports RDP 5.0 (for example, the version that shipped with Windows 2000 Advanced Server)

Setting Up the Server

The following procedures outline the steps necessary to set up your servers to support the terminal server integration. Based on your server's environment, determine which set of steps to follow.

You must match the appropriate files from the installation source to your environment. Otherwise, the extensions will not function properly. If you later install or uninstall the Novell Client, you must modify the SecureLogin modules to match.

Your SecureLogin terminal server components must match the version of SecureLogin you are using. When you upgrade to a new version of SecureLogin, you must also upgrade the integration components.

Copying Protocol Files

WARNING: If you skip this step, Windows will not function properly.

Copy the following files to the Windows system directory (for example, c:\winnt\system32):

- ◆ srv\sl_vc.dll
- ◆ srv\sl_rdp.dll
- ◆ srv\sl_ica.dll

Setting Up GINA

Servers with the Novell Client

- 1** Set up a Novell login extension.

Copy srv\nw\slina.dll to the Windows system directory (for example, c:\winnt\system32).
- 2** Register the login extension.

At the srv\nw directory, double-click Register NT LoginExt.reg.

Servers without the Novell Client

- 1 Replace the server GINA.
Copy `srv\ms\sl_tsgina.dll` to the Windows system directory (for example `c:\winnt\system32`).
- 2 Register GINA.
At the `srv\ms` directory, double-click `winlogon_server.reg`.
- 3 Reboot the server.

Configuring OnDemand

If you have set up a Microsoft Terminal Server with Novell ZENworks® OnDemand Services™ installed, you don't need to install any new components for SecureLogin. OnDemand relies on the DeFrame™ ICA or RDP plug-ins as the client. No workstation components are necessary. When a user authenticates to the Citrix session, Novell SecureLogin launches.

If you use the SecretStore option with OnDemand Dynamic User Creation, make the following changes to the `EnableUserProfileDirectory` value in the `HKEY_LOCAL_MACHINE\SOFTWARE\NOVELL\NICI` registry key:

Value	Type	Description
<code>EnableUserProfileDirectory</code>	DWORD	NICI user files are created in the Application Data\Novell\NICI directory in the user's profile directory

The NICI installation program does not create `EnableUserProfileDirectory`. Therefore, this value is disabled.

NOTE: If the user profile directory is enabled, NICI does not set the Access Control Lists (ACLs) on this directory. NICI relies on the existing security properties (ACLs, inheritance, and ownership) of the user's profile directory.

To configure a DeFrame application object to launch Internet Explorer, with Internet Explorer using the ICA protocol:

- 1 In ConsoleOne®, right-click the Application object.
- 2 Select DeFrame, then click Application Setup.
- 3 Add `prolauncher.exe`.
Enclose `path\applicationname` in quotation marks (for example, "`c:\Program Files\Novell\SecureLogin\prolauncher.exe`" "`c:\Program Files\Internet Explorer\iexplore.exe`").
- 4 Install the SecureLogin client at the Citrix/DeFrame server.

Setting Up Workstations

The following procedures outline the steps necessary to set up your workstations to support the Citrix integration. Based on your client workstation environment, determine which set of steps to follow.

You must match the appropriate files from the installation source to your environment. Otherwise, the extensions will not function properly. If you later install or uninstall the Novell Client or NMAS client, you must modify the SecureLogin modules to match.

Your SecureLogin terminal server components must match the version of SecureLogin you are using. When you upgrade to a new version of SecureLogin, you must also upgrade the integration components.

Your client configuration doesn't need to match your server configuration. For example, you can use a client that has the Novell Client installed and connect to a terminal server that does not (or vice-versa).

Novell Client (without the NMAS Client)

- 1 Set up the Novell login extension.

Copy `wks\nw\slina.dll` to the Windows system directory (for example, `c:\winnt\system32` for Windows NT or `c:\windows\system` for Windows 9x).

The `slina.dll` file is a login extension. After you copy the file, you must register it by using the registry (REG) file.

- 2 Register the login extension.

If you are running Windows NT, Windows XP, or Windows 2000, double-click `Register NT LoginExt.reg`, found in the `WKS\NW` directory.

If you are running Windows 95 or Windows 98, double-click `Register 98 LoginExt.reg`, found in the `wks\nw` directory.

- 3 Set up Microsoft Layer for Unicode on Windows 95/98/ME.

If you are running Windows 9x/ME, copy `redistributable\unicows.dll` to your system directory (for example, `c:\windows\system`).

- 4 Reboot the workstation.

Novell Client (with the NMAS Client)

- 1 Copy `wks\nw\slnmas.dll` to the Windows system directory (for example, `c:\winnt\system32` for Windows NT or `c:\windows\system` for Windows 9x).

The `slnmas.dll` file is not a login extension. Instead, it is called by the NMAS client. If you are using the NMAS client and `slnmas.dll`, it isn't necessary to run the registry (REG) file. You will need to install the version of NMAS client that comes with NSL 3.0.1 or later, which is `slnmas.dll` aware.

- 2 Set up Microsoft Layer for Unicode on Windows 95/98/ME.

If you are running Windows 9x/ME, copy `redistributable\unicows.dll` to your system directory (for example, `c:\windows\system`).

- 3 Reboot the workstation.

Microsoft Workstation with No Novell Client Installed

- 1 Replace the workstation GINA.

Copy `wks\ms\sl_tsc.gina.dll` to the Windows system directory (for example, `c:\winnt\system32`).

- 2 Register GINA.

Double-click `wks\ms\winlogon_client.reg`.

- 3 Reboot the workstation.

NOTE: Windows 95 does not support the GINA credential pass-through without the Novell Client installed.

Virtual Channel Driver

Install the virtual channel driver on workstations, not on servers.

Workstations with the Citrix Client (ICA)

- 1 Install the SecureLogin Citrix ICA virtual channel driver.

Copy `vcd\ica\vdsslsson.dll` to the ICA Client directory (for example, `C:\Program Files\CITRIX\ICA Client`).

- 2 Register the SecureLogin Citrix ICA virtual channel driver.

Make the following changes to the `module.ini` file located in the directory on the client workstation where the ICA Client directory is installed:

- ◆ Add the name of the virtual driver to the end of Virtual Driver line. This line is in the `[ICA 3.0]` section. For example, add

```
, SLSSO
```

- ◆ At the end of the `[VirtualDriver]` section, add a driver assignment statement. For the SLSSO driver, add

```
SLSSO =
```

The extra spaces are for appropriate indentation and are not required.

- ◆ Create a new section, `[SLSSO]` as follows:

```
[SLSSO]
DriverNameWin32 = VDSLSSON.DLL
```

The `vcd\ica` directory has an example `module.ini` file that you can refer to.

- 3 Set up Microsoft Layer for Unicode on Windows 95/98/ME.

If you are running on Windows 9x/ME, copy `redistributable\unicows.dll` to your ICA Client directory (for example, `C:\Program Files\CITRIX\ICA Client`).

Workstations with the Terminal Server Client (RDP)

- 1 Install the SecureLogin Terminal Server virtual channel driver by copying `vcd\rdp\tsslso.dll` to the Windows system directory (for example, `c:\winnt\system32`).

- 2 Register SecureLogin Terminal Server Virtual Channel Driver by double-clicking `VCD\RDP\Terminal Server Driver registration on Client workstation.reg`.

IMPORTANT: This is a per-user setting.

Registry Settings

This section describes optional registry settings that you can make to customize SecureLogin terminal server features.

NOTE: All registry values specified are of string type (REG_SZ).

Auto-Detecting the Client Protocol

To disable the Auto Detection feature, add the following entry to the registry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\Virtual Channel]
"AutoDetect" = "0"
```

To specify the protocol that the server should use, add one of the following entries in the same key:

```
"Protocol" = "RDP"
"Protocol" = "ICA"
```

If the protocol is not specified, the software checks for the presence of ICA. If the ICA protocol is present, the software loads the ICA protocol. Otherwise, the server uses the RDP protocol.

Servers with a Novell Client

To populate a user's common name to the NT Username field during a session login, set the following registry value on the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin\Virtual
Channel\Login\slina]
"PopulateToNT" = "1"
```

Workstations with Terminal Server Client (RDP)

Register SecureLogin Terminal Server Virtual Channel Driver by double-clicking vcd\rdp\Terminal Server Driver registration on Client workstation.reg.

Do this for each user setting.

Localized Machine

To support international versions of Windows, you need to add a localized login window caption to the following registry entry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]
"LogonWindowCaption" = "localized caption"
```

Third-Party GINA

When using the third-party GINA (for example, Citrix GINA), enter the GINA name as follows:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]
"ProtocomPassThruDLL" = "Gina DLL name"
```

If the third-party GINA is using a different login window caption than Microsoft GINA does, enter it as follows in the same key:

```
"LogonWindowCaption" = "Logon window caption"
```

If the Control IDs of the third-party GINA are not the same as Microsoft GINA, enter them as follows:

Create a key as follows:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\Winlogon\ProtocomPassThru]
```

```
"UsernameCtrlID" = "User Name field ID"
```

```
"PasswordCtrlID" = "Password Field ID"
```

```
"DomainCtrlID" = "Domain Name Ctrl ID"
```

```
"IDOK" = "OK Button ID"
```

NOTE: Define Domain Name in a combo box.

Debugging Options

To turn on debugging, double-click the Virtual Channel SSO Debugging Switches.reg file on the workstation or the server.

To view the log file for various components, refer to the following table:

Platform	.DLL File	Path and Log File
Server	slina.dll	c:\winnt\system32\slina.ica.log or c:\winnt\system32\slina.ts.log
Server	sl_tsgina.dll	c:\winnt\system32\sl_tsgina.ica.log or sl_tsgina.ts.log
Workstation	slina.dll	c:\winnt\system32\slina.log
Workstation	sl_tscgina.dll	c:\winnt\system32\sl_tscgina.log
Workstation	vdslssoN.dll	C:\program Files\Citrix\ICA Client\vdslsso.log
Workstation	tsslso.dll	c:\winnt\system32\tsslso.log

To turn debugging off, set "debug" = "0" for each desired component in the registry.

Integrating with Citrix-Published Applications

The SecureLogin ProLauncher.Exe utility starts the SecureLogin components (combroker.exe and proto.exe) and then launches the desired published application with single sign-on support. To launch ProLauncher.Exe if you are using Citrix-published applications, you must modify the command line for each published application.

When the application is closed, ProLauncher terminates the proto.exe or combroker.exe session. That way, these utilities don't leave the Citrix session connected.

ProLauncher must be used with any published applications running on the Citrix server. If ProLauncher is not found within the server's path environment variable, you must include the full path to ProLauncher. For example, replace the command line of the published application as follows:

Before	After
C:\Progra~1\Protocom\SecureLogin\tlaunch.exe /q /auto /eWallData Rumba /pProtocomMainframe	ProLauncher.exe C:\Progra~1\Protocom\SecureLogin\tlaunch.exe /q /auto /e "WallData Rumba" /pProtocomMainframe

Using ProLauncher Syntax

To run ProLauncher, use the following syntax:

ProLauncher [Optional Parameters] executable to run [optional executable's parameters]

IMPORTANT: If your executable contains a path or command line parameters that include spaces, be sure to embed them in quotes. Even if your application normally accepts the parameters with spaces, ProLauncher will interpret them as separate parameters, and unexpected results may occur.

ProLauncher includes two command line parameters that control its behavior:

Parameter	Explanation
/w <exename>	Specifies another process to wait for before closing SecureLogin. Examples: <ul style="list-style-type: none"> ProLauncher.exe /w rumbadsp.exe C:\Progra~1\Protocom\SecureLogin\tlaunch.exe /q /auto /e "WallData Rumba" /pProtocomMainframe ProLauncher.exe /w mspaint.exe run_MSPaint.CMD
/d	Debug option. This generates a debug log file (c:\prolauncher.log) and shows dialog boxes during the progress of ProLauncher. The switch must appear before the executable to run. Examples: <ul style="list-style-type: none"> ProLauncher.exe /w rumbadsp.exe /d C:\Progra~1\Protocom\SecureLogin\tlaunch.exe /q /auto /e "WallData Rumba" /pProtocomMainframe ProLauncher.exe /w /d mspaint.exe run_MSPaint.CMD

8

Setting Up Terminal Emulation

This section contains information on the following:

- ◆ [“Overview” on page 117](#)
- ◆ [“Setting Up Terminal Emulation” on page 118](#)
- ◆ [“Setting Up a Shortcut” on page 136](#)
- ◆ [“Using Command Line Parameters” on page 137](#)
- ◆ [“Configuring Backup Sessions” on page 138](#)
- ◆ [“Determining Which Session File To Automatically Use” on page 139](#)
- ◆ [“Using Terminal Launcher With Non-HLLAPI-Compliant Emulators” on page 139](#)

Overview

Terminal Launcher, a component of SecureLogin, enables users to log in to any type of host that requires a login using an emulator (for example, ACF2 or RACF mainframe, a UNIX host, or a Cisco router).

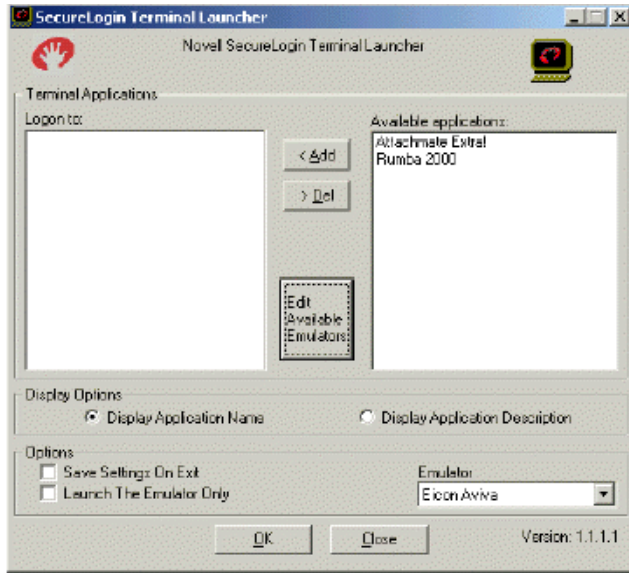
Terminal Launcher supports the following emulator types:

- ◆ HLLAPI
- ◆ DDE
- ◆ VBA
- ◆ Generic
- ◆ Advanced Generic

Terminal Launcher does the following:

- ◆ Acts as the translator between an emulator login sequence and a user’s variables (for example, the username and password) stored in SecureLogin.
- ◆ Coordinates the information being entered onto a mainframe or Telnet screen.

The following figure illustrates Terminal Launcher’s main window.



Terminal Launcher can use High Level Language Application Programming Interface (HLLAPI) commands to interface with a wide range of mainframe emulators.

The SecureLogin scripting language enables you to enter a variety of keystrokes through Terminal Launcher to an emulator. For information on scripting and the commands used with Terminal Launcher, see the [Script Commands](#) guide.

To launch Terminal Launcher, click Start > Programs > Novell SecureLogin > Terminal Launcher.

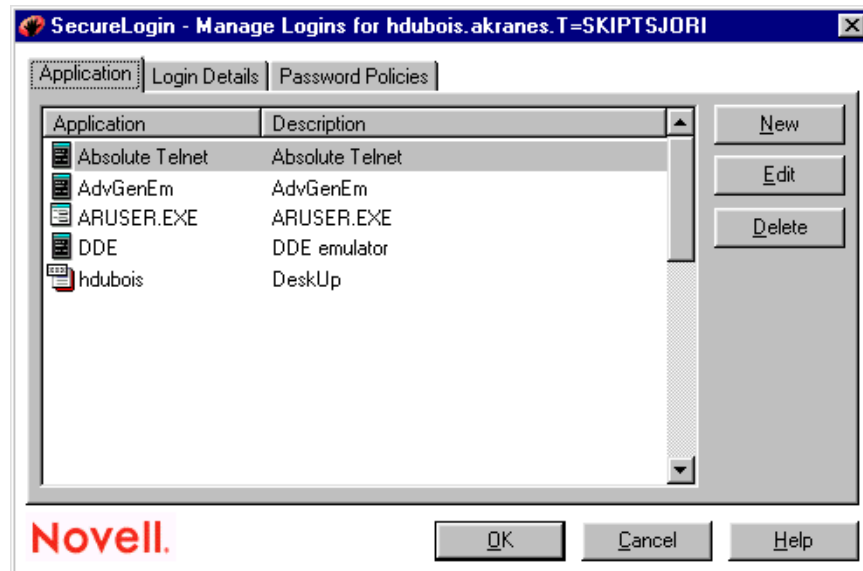
Setting Up Terminal Emulation

To set up terminal emulation, create a script for the emulator, specifying Terminal Launcher as the script type. Then configure the emulator and create a login.

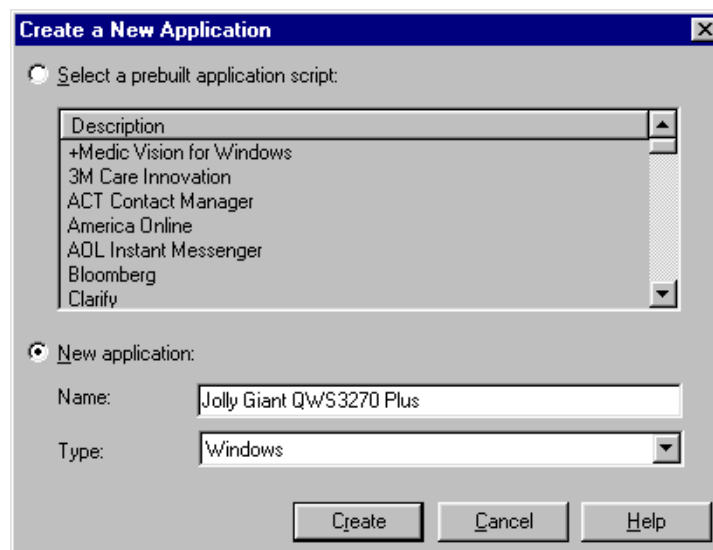
Creating an Emulator Script

The following example sets up SecureLogin Terminal Launcher to single sign-on to a session using Jolly Giant QWS3270 Plus.

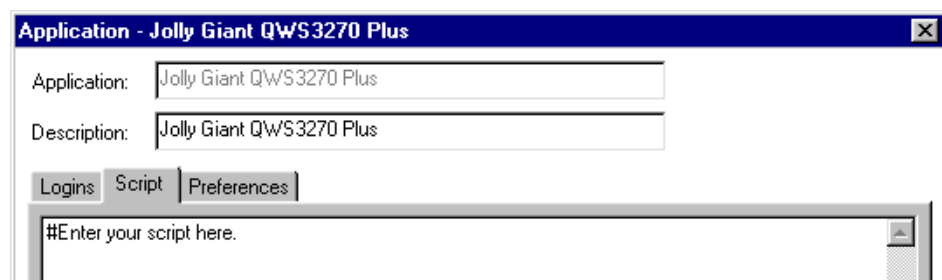
- 1 Double-click the SecureLogin icon on the system tray, then click Manage Logins.
- 2 Click Application, then click New.



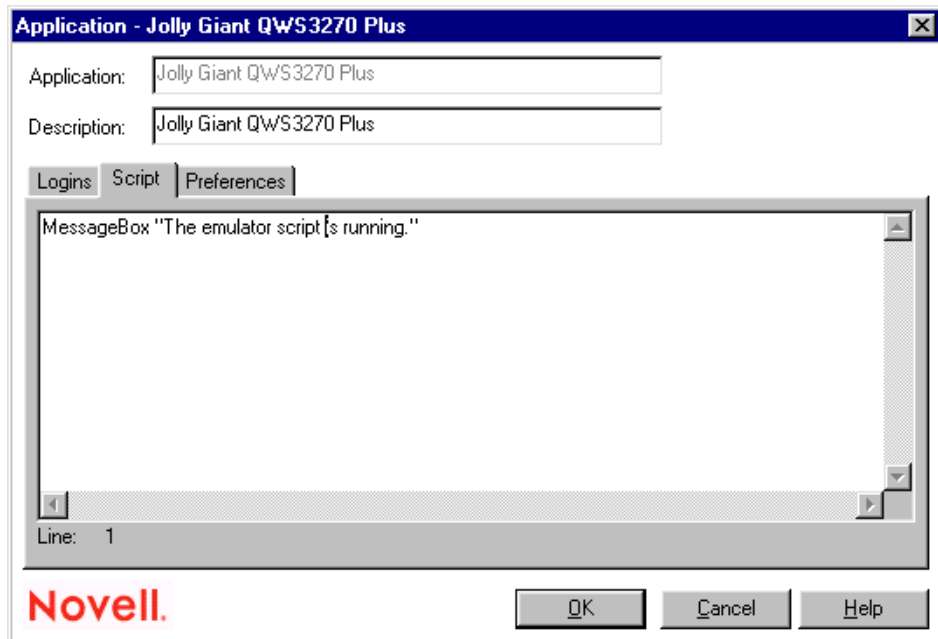
- 3 Select New Application, type a name in the Name text box, select Terminal Launcher as the type, then click Create.



- 4 Click Script.



- 5 Type a single command, then click OK.



For example, type a MessageBox command. By entering one command, you can find out the following:

- ◆ Terminal Launcher is working as expected.
- ◆ The script is ready for you to enter the appropriate commands.

6 Save the data and close open windows by clicking OK.

Configuring Emulators

SecureLogin Terminal Launcher includes configurations for the following emulators:

Emulator	Emulator
Attachmate* Extra*	NetTerm v4.2
Attachmate Extra 2000	NS/Elite
Attachmate KEA!	PASSPORT* TN 3270E
Chameleon* Hostlink	QVT
Eicon* Aviva*	QWS3270 Plus
GLink	SDI* TN3270
HBO Star Navigator	TeraTermPro
IBM* Personal Communications	TLaunch Options
IDXTerm Healthcare	ViewNow
Info Connect	Wall Data RUMBA*
Microsoft* Telnet 2000	Wall Data RUMBA 2000

Emulator	Emulator
Microsoft Telnet NT	Wall Data RUMBA Web To Host
Microsoft Telnet Win 9x	Windows Telnet VT
Mocha W32 Telnet	WRQ* Reflection*

You can configure Terminal Launcher to do the following:

- ◆ Connect to the mainframe or host, wait for the login sequence, then enter usernames and passwords.
- ◆ Work with a range of different terminal emulators.
- ◆ Navigate to a particular subsystem or menu within an application.
- ◆ Incorporate into SecureLogin scripts any keystrokes that an application accepts, so that Terminal Launcher can send the keystrokes to the host.

At the corporate level, one script can accommodate all users. SecureLogin applies user-specific variables.

To configure Terminal Launcher you must specify any required mainframe session file and the path to the emulator executable.

Configuring WinHLLAPI, HLLAPI, or 16-Bit HLLAPI Emulators

This section can help you configure WinHLLAPI, HLLAPI, or 16-bit HLLAPI emulators. If you select the wrong HLLAPI type, however, Terminal Launcher will fail. To find out the HLLAPI type, do one of the following:

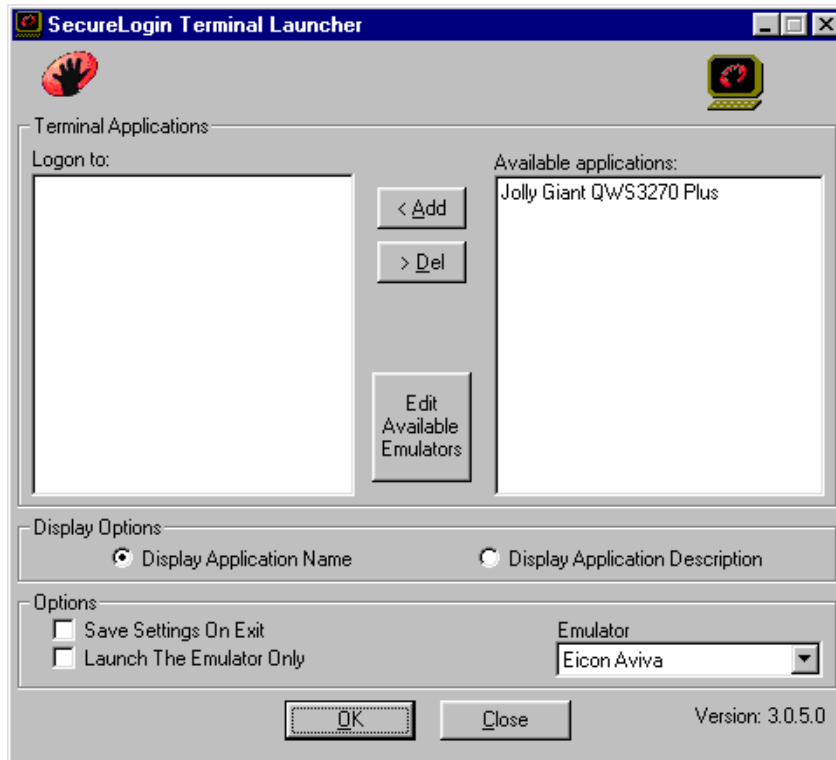
- ◆ Consult the documentation on the emulator to find out the following:
 - ◆ Whether the emulator supports HLLAPI
 - ◆ The type of HLLAPI that the emulator supports (WinHLLAPI, HLLAPI, or 16-bit HLLAPI)
- ◆ Check the .dll files by using Dependency Walker.

NOTE: Dependency Walker won't open 16-bit .dll files.
- ◆ Create a configuration for each of the three HLLAPI types.

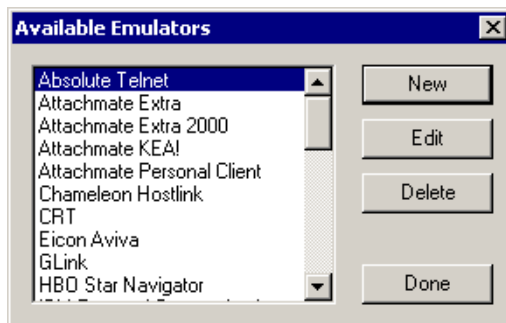
HINT: Some emulators require that you configure HLLAPI session names and the configuration within the session files or the main application. Otherwise, HLLAPI won't work. Even though you might have configured everything with SecureLogin, HLLAPI might still fail due to unconfigured emulators.

- 1 Click Start > Programs > Novell SecureLogin > Terminal Launcher.

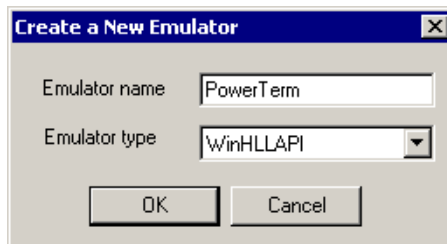
As the following figure illustrates, Terminal Launcher displays the application that you created the script for:



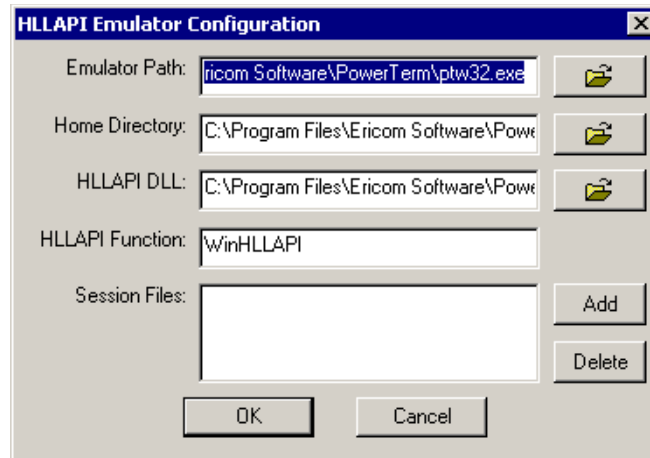
- 2 Click Edit Available Emulators > New.



- 3 Type a name for the emulator, select WinHLLAPI, HLLAPI, or HLLAPI16 as the emulator type, then click OK.



- 4 Type values, then click OK.



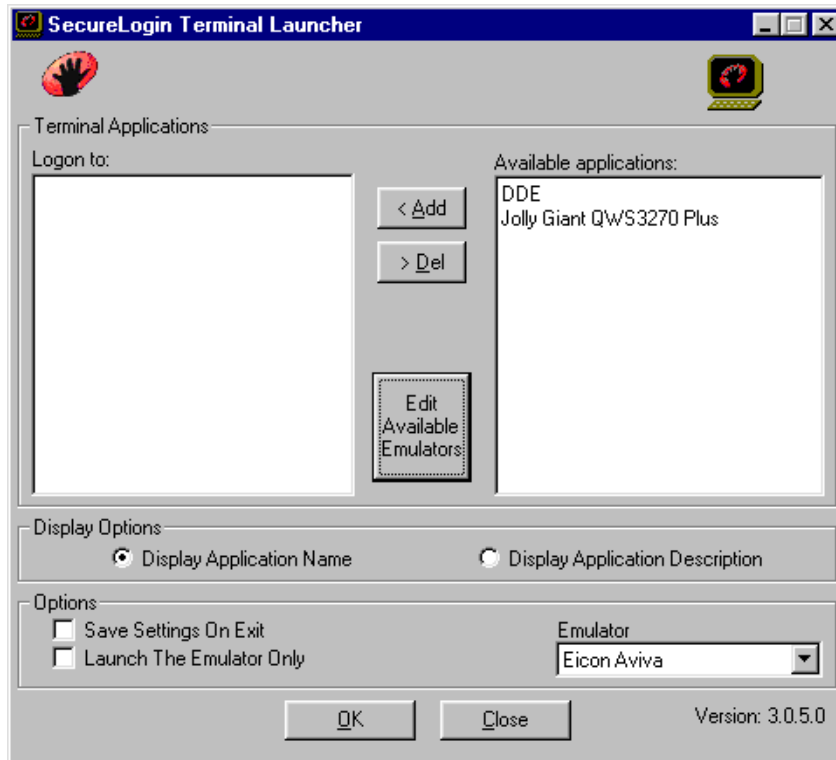
Field	Description
Emulator Path	The directory path and executable filename of the emulator. Either type the path or use the Browse button located to the right of the text box. You can type short (8.3 format) or long filenames. Enclose long filenames in quotes (for example, "c:\Program Files\emulator").
Home Directory	The directory path to files for this emulator.
HLLAPI.DLL	The directory path to the .dll file for this emulator, along with the .dll filename. This file is in the Home directory. Typically, the filename contains "hllapi" in the name, but on occasion you need to refer to the documentation on the emulator. Validate the .dll file by using Dependency Walker (depends.exe), which is available from the Dependency Walker Web site (http://www.dependencywalker.com) .
HLLAPI Function	The name of the HLLAPI function contained within the hllapi.dll file. Find or validate this information by using Dependency Walker. The function is case sensitive. Enter the function name exactly as you find it in Dependency Walker.
Session Files	The session files for the emulator. Type the path and session file name. Enclose long names within quotes (for example, "C:\Program Files\Sessions\Session1.xxx").

- 5** In the Available Emulators dialog box, click Done.

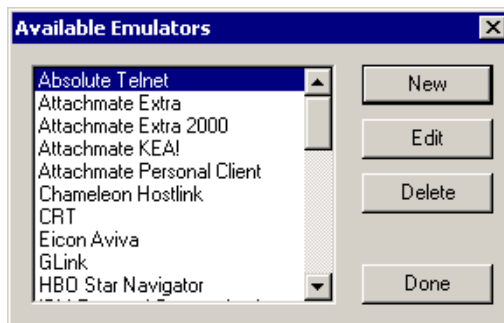
Configuring a DDE Emulator

- 1** Click Start > Programs > Novell SecureLogin > Terminal Launcher.

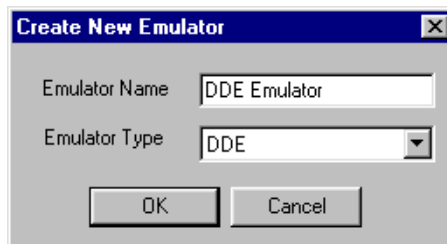
As the following figure illustrates, Terminal Launcher displays the application that you created the script for:



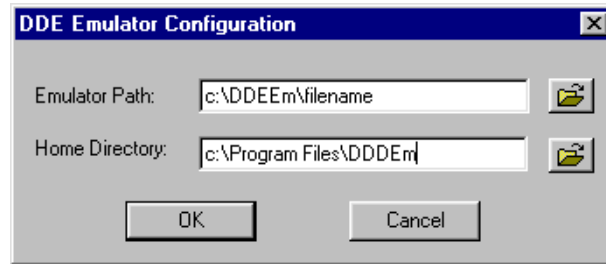
2 Click Edit Available Emulators > New.



3 Type a name for the emulator, select DDE as the emulator type, then click OK.



4 Type values, then click OK.



Field	Description
Emulator Path	The directory path and executable filename of the emulator. Either type the path or use the Browse button located to the right of the text box. You can type short (8.3 format) or long filenames. Enclose long filenames in quotes (for example, "c:\Program Files\emulator").
Home Directory	The directory path to where the executable is located.

- 5 In the Available Emulators dialog box, click Done.

Configuring a VBA Emulator

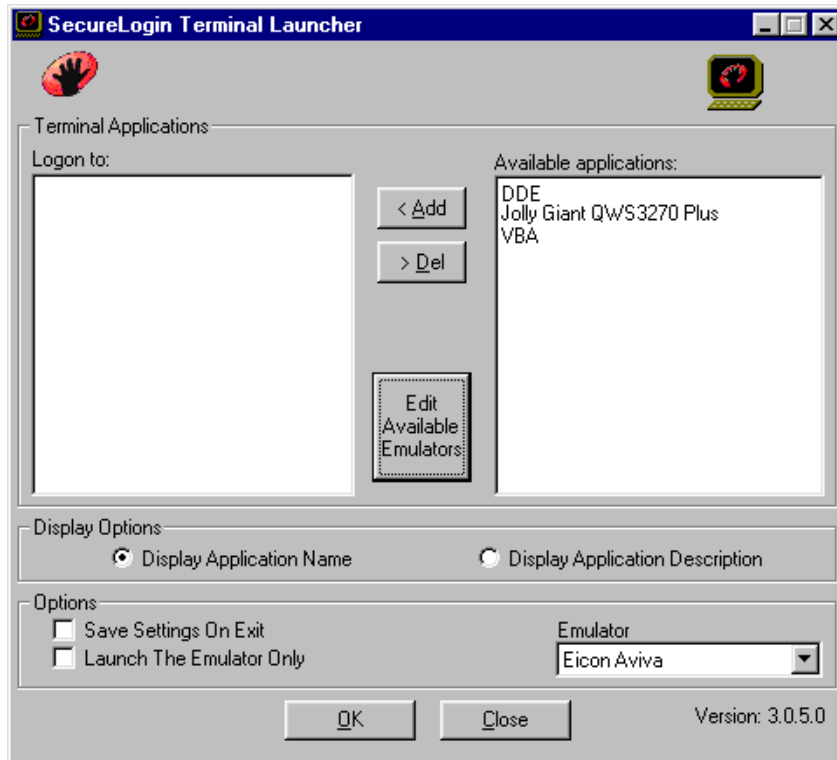
VBA emulators support the Visual Basic* scripting language. In most cases, it is possible to write a macro for the emulator. The macro prompts SecureLogin to enter credentials.

Configuring VBA emulators to work with Terminal Launcher is a specialized field and is specific to each emulator.

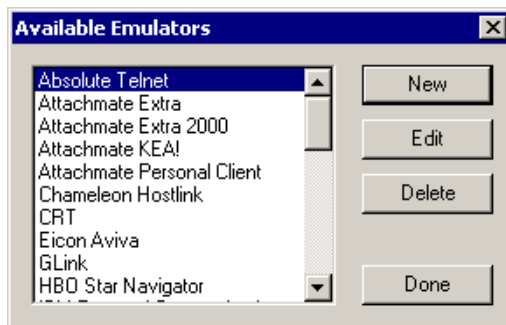
VBA emulators can often be configured as Generic emulators. However, generic configuration offers limited functionality.

- 1 Click Start > Programs > Novell SecureLogin > Terminal Launcher.

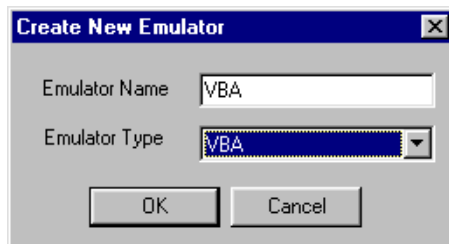
As the following figure illustrates, Terminal Launcher displays the application that you created the script for:



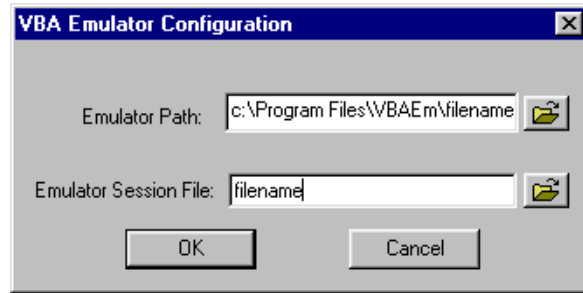
2 Click Edit Available Emulators > New.



3 Type a name for the emulator, select VBA as the emulator type, then click OK



4 Type values, then click OK.



Field	Description
Emulator Path	The directory path and executable filename of the emulator. Either type the path or use the Browse button located to the right of the text box. You can type short (8.3 format) or long filenames. Enclose long filenames in quotes (for example, "c:\Program Files\emulator").
Emulator Session	The session file for the emulator. Type the path and session file name. Enclose the path and filename within quotes (for example, "C:\Program Files\Sessions\Session1.xxx").

- 5 In the Available Emulators dialog box, click Done.

Configuring a Generic Emulator

The Generic Emulator option enables you to configure SecureLogin's Terminal Launcher to interface with emulators that do not provide HLLAPI, VBA, or DDE support.. Terminal Launcher interfaces with generic emulators though the Windows Clipboard by copying and pasting.

Because an emulator doesn't have to be programmed to allow external programs to interface with it, almost any emulator can be configured as a generic emulator.

Normal generic emulators have Select All, Copy, and Paste functions. These functions are most often found in the Edit menu, which is at the top of the emulator screen. However, buttons or keyboard shortcuts might be available.

Terminal Launcher uses these functions to interface with the emulator. Upon running a WaitForText command in a script, Terminal Launcher repeatedly simulates the selection of Select All, then Copy, which places the content of the emulator's screen on the Windows Clipboard. Terminal Launcher then searches the content of the Clipboard for the text it is waiting for.

If the text is not found, Terminal Launcher repeats the procedure, which usually takes about half a second.

Also, the emulator screen flickers while the WaitForText command is being executed. This is normal. See [BeginSplashScreen / EndSplashScreen](#) in the *Script Commands* guide.

If Terminal Launcher finds the text it is looking for, Terminal Launcher goes to the next line of the script, which is most often a Type command. The Type command enters a username or password. To enter the text into the emulator, you can configure Terminal Launcher to type the text in, or to use the Clipboard.

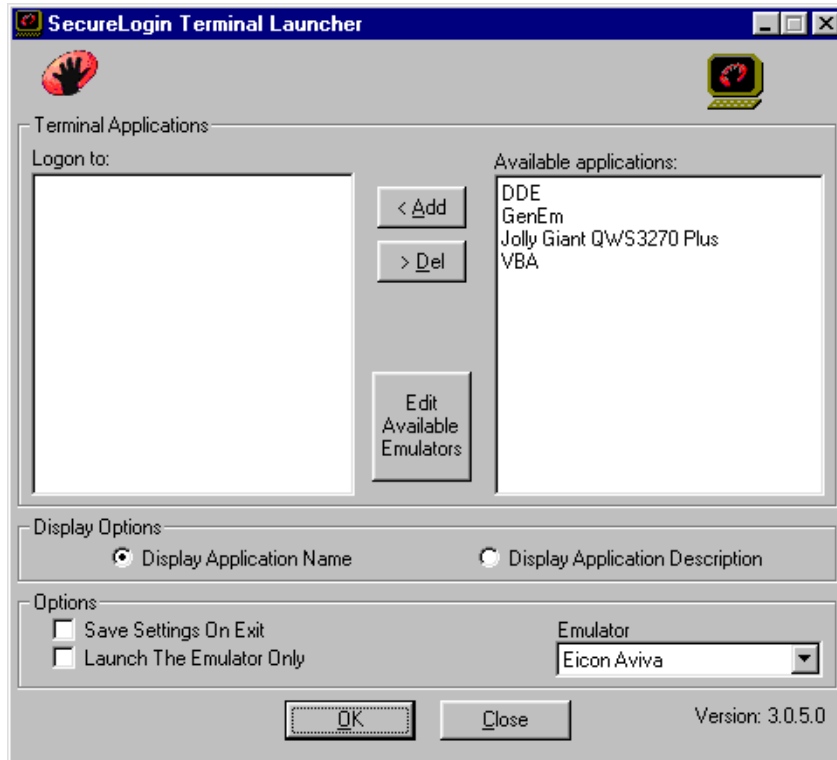
When using the Clipboard, Terminal Launcher copies the text (for example, a username) to the Clipboard and then simulates the selection of the Paste function of the emulator. This procedure

copies the text to the screen of the emulator. When using the direct typing method, Terminal Launcher simulates pressing the applicable keys on the keyboard.

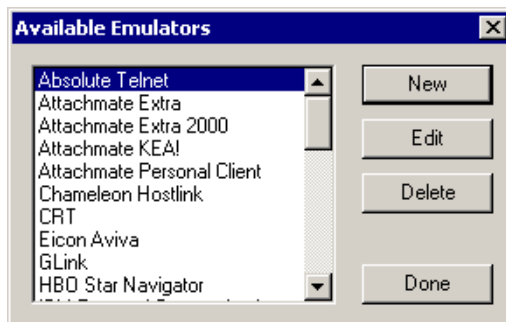
The process is then repeated for the password, and the user's login to the mainframe session is complete.

- 1 Click Start > Programs > Novell SecureLogin > Terminal Launcher.

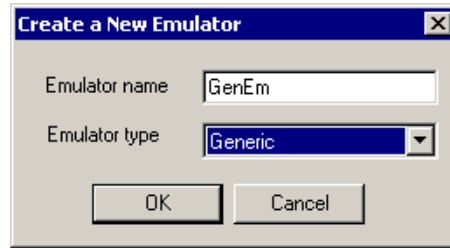
As the following figure illustrates, Terminal Launcher displays the application that you created the script for:



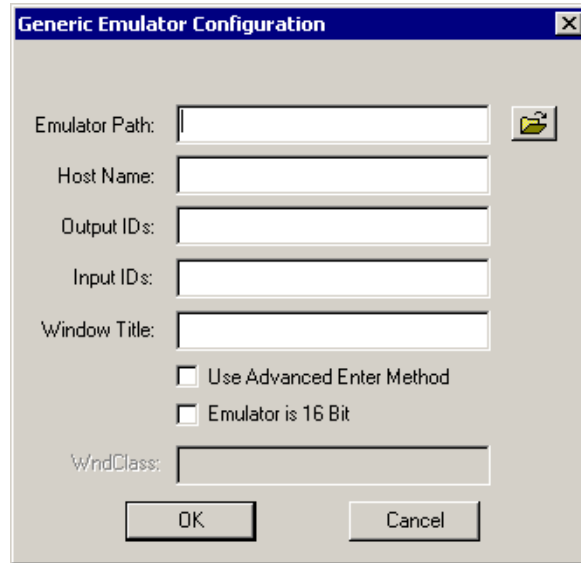
- 2 Click Edit Available Emulators > New.



- 3 Type a name for the emulator, select Generic as the emulator type, then click OK.



4 Type values, then click OK.



Field	Description
Emulator Path	The directory path and executable filename of the emulator. Either type the path or use the Browse button located to the right of the text box. You can type short (8.3 format) or long filenames. Enclose long filenames in quotes (for example, "c:\Program Files\emulator").
Host Name	The IP address, host name, or emulator session file you want Terminal Launcher to connect to or load. Occasionally, emulators require command line switches before they accept connection commands at startup. If so, include the switches in the Host Name text box. Scenario for Including a Command Line Switch: An emulator requires the /h switch so that the emulator can accept an IP address at startup. Henri types /h 192.168.130.222 in the Host Name text box.
Output IDs	The Control ID for the Copy function of the emulator. For information on finding Output IDs, see Appendix A, "Finding Control IDs and Offsets," on page 157 . Some emulators allow a keyboard simulation alternative (for example, CTRL+C) instead.

Field	Description
Input IDs	<p>The Control ID for the Paste function of the emulator. For information on finding Input IDs, see Appendix A, "Finding Control IDs and Offsets," on page 157.</p> <p>Some emulators allow a keyboard simulation alternative (for example, CTRL+C) instead.</p>
Window Title	<p>Assists Terminal Launcher in detecting the emulator window. If no Window Title is specified, Terminal Launcher might not detect the emulator opening.</p> <p>To find the Window Title, use Window Finder. Run Window Finder, then right-click and drag the SecureLogin icon to the title bar of the emulator. The required value will be shown as the second-from-last entry (Window Text) in Window Finder.</p> <p>Some emulators hide the real Window Title.</p> <p>There is no rule to describe which text you should enter into the Terminal Launcher configuration. (It depends on how the emulator hides it.) First, try the configuration without a Window Title specified. Next, try the text that is actually displayed in the title bar of the emulator. Finally, try the real Window Title.</p>
Use Advanced Enter Method	<p>Enables you to use the Advanced Enter method. This method is necessary for Lawson and StarNavigator emulators. Advanced Enter sends the \n character sequence to the emulator for the Type @E command.</p>

- 5 In the Available Emulators dialog box, click Done.

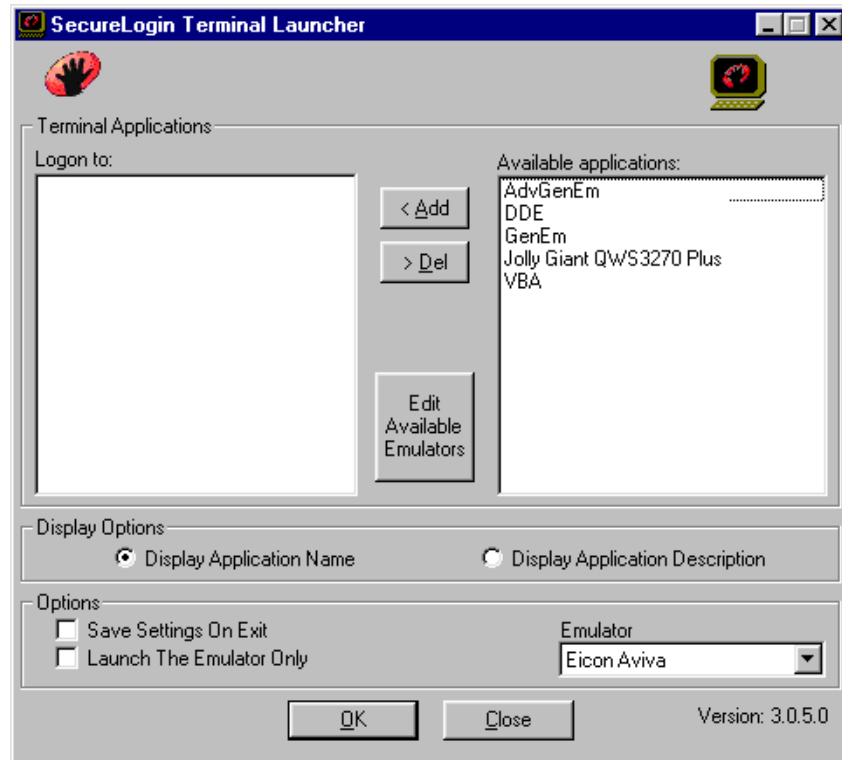
Configuring an Advanced Generic Emulator

Advanced generic emulators have Copy and Paste functions, but do not have a Select All function.

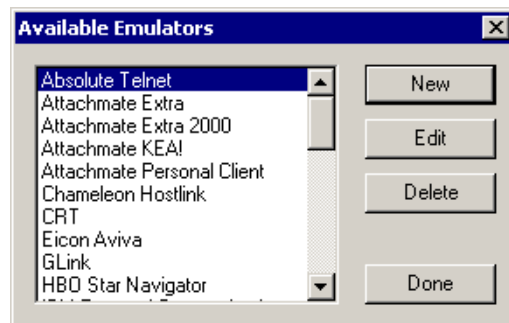
For advanced generic emulators, Terminal Launcher follows the same process as for generic emulators, except for one difference. Instead of simulating the selection of Select All, Terminal Launcher clicks and drags the mouse cursor over the emulator screen. This procedure selects all the text that the emulator is displaying.

- 1 Click Start > Programs > Novell SecureLogin > Terminal Launcher.

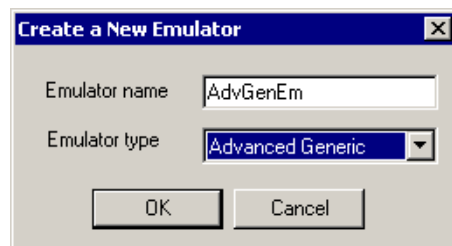
As the following figure illustrates, Terminal Launcher displays the application that you created the script for:



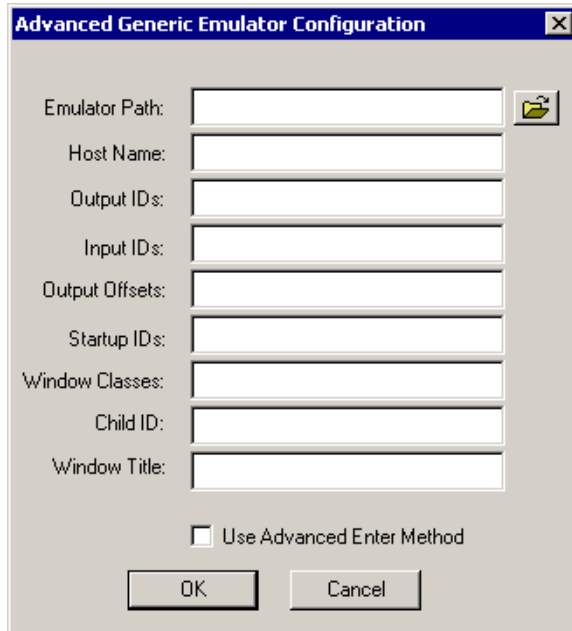
- 2 Click Edit Available Emulators > New.



- 3 Type a name for the emulator, select Advanced Generic as the emulator type, then click OK.



- 4 Type values, then click OK.



Field	Description
Emulator Path	The directory path and executable filename of the emulator. Either type the path or use the Browse button located to the right of the text box. You can type short (8.3 format) or long filenames. Enclose long filenames in quotes (for example, "c:\Program Files\emulator").
Host Name	The IP address, host name, or emulator session file you want Terminal Launcher to connect to or load. Occasionally, emulators require command line switches before they accept connection commands at startup. If so, include the switches in the Host Name text box. Scenario for Including a Command Line Switch: An emulator requires the /h switch so that the emulator can accept an IP address at startup. Henri types /h 192.168.130.222 in the Host Name text box.
Output IDs	The Control ID for the Copy function of the emulator. Some emulators allow a keyboard simulation alternative (for example, CTRL+C) instead. For information on finding Output IDs, see Appendix A, "Finding Control IDs and Offsets," on page 157 .
Input IDs	The Control ID for the Paste function of the emulator. Some emulators allow a keyboard simulation alternative (for example, CTRL+C) instead. For information on finding Input IDs, see Appendix A, "Finding Control IDs and Offsets," on page 157 .

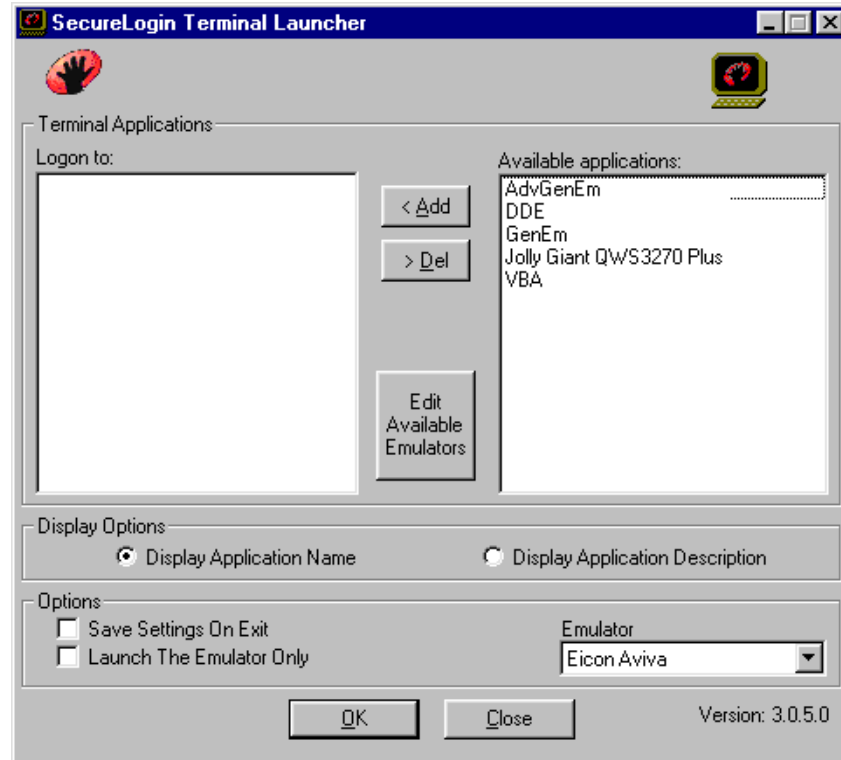
Field	Description
Output Offsets	<p data-bbox="689 157 1449 270">For Terminal Launcher to select the text on the screen without a Select All function, Terminal Launcher needs to use the mouse to copy from the screen. Select all the text by clicking and dragging the cursor over the entire emulator screen.</p> <p data-bbox="689 298 1449 411">For Terminal Launcher to do this correctly, you need to specify how much space to allow for the toolbar and other bars when Terminal Launcher starts to drag. The Output Offset is the specified number, which tells Terminal Launcher where to start the click-and-drag process.</p> <p data-bbox="689 439 1449 552">The Output Offset is a set of two numbers, separated with a comma and no space. The numbers can be anything from zero up to tens of thousands. However, 10,000 is generally the highest number that is needed.</p> <p data-bbox="689 580 1449 606">To find the Output Offsets, you can use nsfinder.exe or trial and error.</p> <p data-bbox="689 635 1449 768">To use nsfinder.exe, download the utility from Novell's Support Web site (http://support.novell.com/servlet/filefinder?name=nsfinder.exe). A text file that explains how to use nsfinder is downloaded along with the utility. TID 2965468 provides the download. Also see "Finding Offsets" on page 165.</p> <p data-bbox="689 796 1449 848">To use the trial-and-error method, start with high numbers (for example: 10000,10000) and work your way down.</p> <p data-bbox="689 876 1449 989">The first number is the horizontal offset, and the second number is the vertical offset. Watch where the mouse starts to click and drag. Then lower the numbers until the mouse clicks in the top lefthand corner of the emulator screen.</p> <p data-bbox="689 1018 1449 1044">Normally, the Output Offset number is around 500,7000 or lower.</p> <p data-bbox="689 1072 1449 1098">Trial and error becomes easier with experience.</p> <p data-bbox="689 1126 1449 1199">IMPORTANT: The offsets for an Advanced Generic emulator depend on screen resolutions. Therefore, an offset for a workstation set to 800 x 600 pixels will differ from a workstation set to 1074 x 768 pixels.</p> <p data-bbox="689 1227 1449 1340">We recommend that you create multiple emulator definitions for each screen resolution. The actual SecureLogin script remains the same, but you deliver a unique Terminal Launcher configuration for each different screen resolution.</p> <p data-bbox="689 1368 1449 1481">WARNING: If you enter the wrong offsets, SecureLogin might select undesired locations on the workstation's desktop. These locations might close, maximize, or minimize other applications, resize the Windows task bar, or perform some other unwanted task.</p>
Startup IDs	<p data-bbox="689 1501 1449 1594">The Control ID numbers of any functions or buttons that you want Terminal Launcher to select before attempting to log in. An example of this is a Connect button if the emulator does not automatically connect</p> <p data-bbox="689 1622 1449 1649">If several functions or buttons are needed, separate them with a comma.</p>

Field	Description
Windows Classes	<p>Assist Terminal Launcher in detecting the emulator window. If no Window Class is specified, Terminal Launcher might not detect the emulator opening.</p> <p>To find the Window Class, use Window Finder. Run Window Finder, then right-click and drag the SecureLogin icon to the title bar of the emulator. The required value is shown as the third-from-last entry (Class Name) in Window Finder.</p>
Child ID	<p>A Child ID is given to each child window that the main application launches.</p> <p>Only enter a Child ID into the TLaunch configuration if it is necessary to interact with a child window.</p>
Window Title	<p>Assists Terminal Launcher in detecting the emulator window. If no Window Title is specified, Terminal Launcher might not detect the emulator opening.</p> <p>To find the Window Title, use Window Finder. Run Window Finder, then right-click and drag the SecureLogin icon to the title bar of the emulator. The required value will be shown as the second-from-last entry (Window Text) in Window Finder.</p> <p>Some emulators hide the real Window Title.</p> <p>There is no rule to describe which text you should enter into the Terminal Launcher configuration. (It depends on how the emulator hides it.) First, try the configuration without a Window Title specified. Next, try the text that is actually displayed in the title bar of the emulator. Finally, try the real Window Title.</p>
Use Advanced Enter Method	<p>Enables you to use the Advanced Enter method. This method is necessary for Lawson and StarNavigator emulators. Advanced Enter sends the \n character sequence to the emulator for the Type @E command.</p>

5 In the Available Emulators dialog box, click Done.

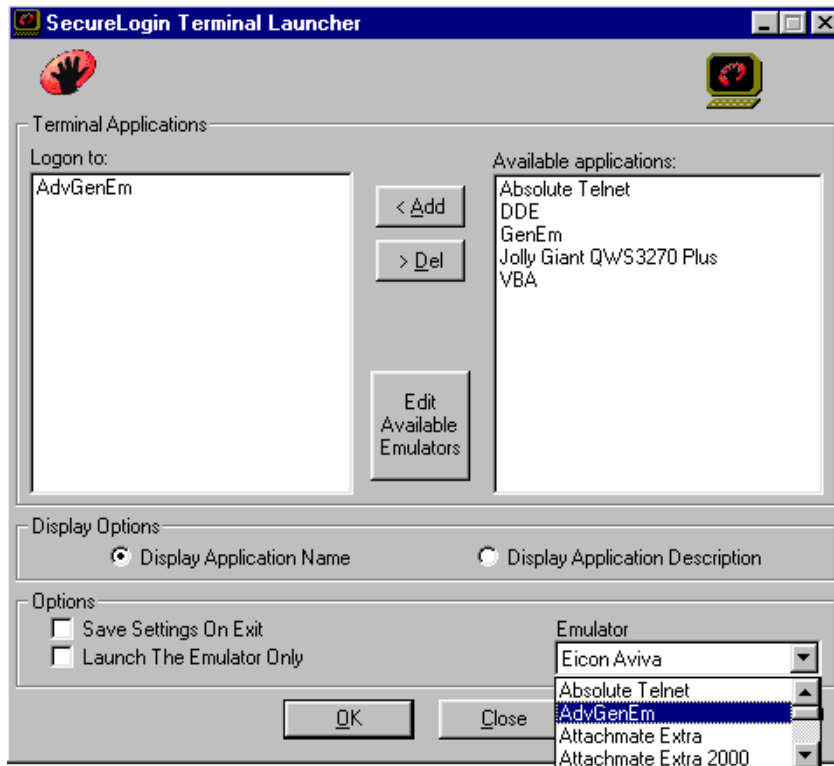
Creating a Login for an Emulator

- 1 From the list in the Available Applications pane, click the application that one you want to log in to, then click Add.



To move an entry from one side to another, you can double-click it.

- 2 Select the emulator from the Emulator drop-down list, then click Launch.



The selected application script runs, using the selected emulator. d

The first time the script is run, you encounter a prompt to enter your Username and Password. Enter the required values, then click OK. Terminal Launcher launches the emulator, enters your username and password, and logs you in to a session.

Setting Up a Shortcut

You can set up a shortcut for the application. The following steps create a Terminal Launcher desktop shortcut called Simple Login. This shortcut launches and logs the user into an Eicon Aviva session.

- 1** Right-click the desktop, select New, then click Shortcut.
- 2** Enter (or browse to) the path for tlaunch.exe.
For example, enter "c:\Program Files\novell\securelogin\tlaunch.exe". Include quotation marks.
- 3** To the end of this line, add `/auto/p[application_name]e[emulator_name]`.

For example, enter

```
/auto/pSimple Login/eEicon Aviva
```

/auto: Enables the command line facility of Terminal Launcher. Include /auto in the command line so that other command line options will work.

/p: Runs the specified application script. The name of the application script follows this parameter. Enter the application name exactly as it appears in the Available Applications list in Terminal Launcher.

/e: Launches the specified emulator. Enter the emulator name exactly as it is listed in the Terminal Launcher Emulator drop-down list.

The SecureLogin Terminal Launcher can launch up to 15 applications at a time, as long as you have enough sessions defined for the particular emulator you are using. To open several applications at once, add one more **/papplication_name** for each additional application.

The shortcut box contains this shortcut line:

```
"C:\Program Files\Novell\Securelogin\Tlaunch.exe" /auto/pSimple  
Login /eEicon Aviva
```

- 4 Click Next, name the new application shortcut, then click Finish.

When you double-click the shortcut, the shortcut launches Eicon Aviva and runs the selected script.

Using Command Line Parameters

Terminal Launcher can use the following command line parameters.

Parameter	Description
<i>/auto</i>	Tells Terminal Launcher that you are running the command line version. This parameter must be in the command line for the other command line options to work.
<i>/b</i>	Specifies background authentication mode.
<i>/eemulator_name</i>	Launches the specified emulator.
<i>/hllapi_short_name</i>	Forces Terminal Launcher to connect to the specified HLLAPI session.
<i>/l</i>	Instructs Terminal Launcher to not launch the emulator. Terminal Launcher tries to run the script, assuming that the emulator has already been launched.
<i>/kexecutable_name</i>	Kills the specified executable before launching an emulator.
<i>/m</i>	Allows multiple (sequential) connections to particular sessions.
<i>/n</i>	Launches the selected emulator without running a script (equivalent to the Emulator Only check box in the main program).
<i>/nnumber_1-15</i>	Launches the specified number of sessions without running scripts.
<i>/papplication_name</i>	Runs the specified application.
<i>/q</i>	Specifies Quiet Mode (no cancel dialog).
<i>/s</i>	Suppresses errors.
<i>/t</i>	Enables unlimited timeout when connecting to an emulator.
<i>/wprocess name</i>	Waits until the specified process name is running before running the script.
<i>/xparameters</i>	Sets HLLAPI session parameters before doing the HLLAPI connect. Not for general use.

The following examples include parameters that Terminal Launcher uses:

- ◆ `Tlaunch.exe /auto`

This example creates a shortcut that launches the SecureLogin Terminal Launcher.

- ◆ `Tlaunch.exe /auto /eEicon Aviva /pApplication1 /pApplication2`

This example creates a shortcut that launches two SecureLogin Terminal Launcher application scripts. The scripts launch two sessions of Eicon Aviva, one named Application1, the other named Application2.

- ◆ `Tlaunch.exe /auto /pTSO`

This example creates a shortcut that launches the SecureLogin Terminal Launcher application script named TSO.

- ◆ `Tlaunch.exe /auto /n3`

This example creates a shortcut that launches the SecureLogin Terminal Launcher and opens three sessions. The three sessions are whichever emulator was last used with the Launch The Emulator Only option selected and then closed with the Save Settings On Exit option selected.

When an emulator or application is not specified on the command line, Terminal Launcher uses the settings stored in the user settings file (`usersettings.ini`). You can modify these settings.

- 1 Check the Save Settings on Exit check box.
- 2 Close the main program.

Configuring Backup Sessions

Each Terminal Emulator that is configured must have a number of backup sessions configured for it. For most emulators, you are required to have one session file for every session that you want to have running at the same time. These are usually stored as separate files.

When you configure an emulator for use with Terminal Launcher, you must input a session file for it to use. To tell Terminal Launcher that it can use more than one session file, complete the following steps:

- 1 Launch Terminal Launcher, then click Edit Available Emulators.
- 2 Select the correct emulator from the Available Emulators window, then click Edit.
- 3 Add the backup session files to the Session Files dialog box.

You can launch only as many emulator sessions as there are session files defined.

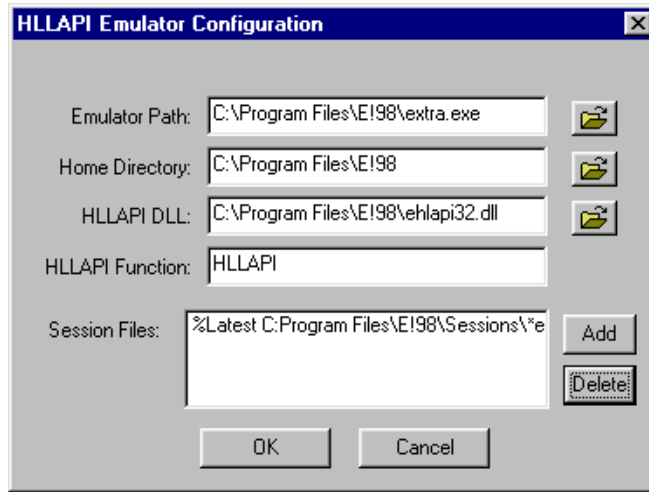
NOTE: After the emulator is launched, these session files will be executed as a command line parameter. Some emulators (such as QWS3270 Plus) do not have session files. Instead, these emulators have individual sessions stored in the registry. Think of these session files as command line parameters that will be passed to the executable.

Determining Which Session File To Automatically Use

A session file tells the emulator how to connect to the mainframe. In some environments, and with emulators like Attachmate Extra, users on the network might have given their session file a unique name. This means that Terminal Launcher might need to be configured individually for each user.

Terminal Launcher includes an option that allows it to determine the last-used session file and start that mainframe configuration. This option reduces or eliminates the need to manually configure each user's environment when unique session filenames are used.

To configure Terminal Launcher to use the last-used configuration file, use the command %Latest in the Session Files section of Terminal Launcher. The following figure illustrates this option:



This example entry causes Terminal Launcher to search the E!98\Sessions directory, looking for the .EDP file with the newest date and time. Terminal Launcher then launches that file with the emulator and connects to the mainframe.

If the file mainframe.edp had the most recent date and time in the Sessions directory, the command line would look like the following:

```
C:\Program Files\E!98\extra.exe c:\Program Files\E!98\Sessions\mainframe.edp
```

Using Terminal Launcher With Non-HLLAPI-Compliant Emulators

You can use Terminal Launcher with Terminal Emulators that do not support HLLAPI but do support scripting that is able to call external .dll files. To do this, you must create a script that asks SecureLogin for commands one at a time and then interprets the commands received.

Configuring Reflection 8 for UNIX and Digital

The following script has been tested with Reflection 8 for UNIX and Digital.

```
Sub SecureLogin()  
  Dim SecureLoginObject As ISLBroker  
  Dim ReturnCode As Long  
  Dim Data As String  
  Dim targ As Long  
  Dim FunctType As Long
```

```

Dim CR As String
Dim temp As String

Session.Wait 0.1'The waits are necessary for the screen to be updated.

Set SecureLoginObject = New SLBroker
CR = Chr$(rcCR) ' Chr$(rcCR) = Chr$(13) = Control-M
SecureLoginObject.LoadScript
While (1 = 1)
  FunctType = 0
  'retrieve command from VBABork
  SecureLoginObject.GetCommand FunctType, targ, Data
  If FunctType = SecureLoginObject.SetCursor Then
    ' SetCursor is not supported
    ReturnCode = 0
  ElseIf FunctType = SecureLoginObject.TypeText Then
    If (StrComp(Data, "@E", vbTextCompare) = 0) Then
      Session.Transmit CR
    Else
      Session.Transmit Data
    End If
    Session.Wait 0.1
    ReturnCode = 0
  ElseIf FunctType = SecureLoginObject.ScanForText Then
    bResult = Session.FindText(Data, Session.ScreenTopRow, 0)
    ReturnCode = 0
    If bResult = True Then
      ReturnCode = 1
    End If
  Else
    ' End of script
    GoTo ErrorHandler
  End If
  SecureLoginObject.SetReturnCode ReturnCode
Wend
ErrorHandler:
End Sub

```

The script should also work for Reflection 7. Reflection 6 and earlier versions require a different Reflection script because these versions use Reflection Basic instead of VBA (Visual Basic for Applications). If you require a script for Reflection 5.21, contact [Novell Support \(http://support.novell.com\)](http://support.novell.com).

To use Reflection for UNIX and Digital, you must add the Sub SecureLogin() script by using the macro editor in Reflection. In addition to adding this macro, you must go to the macro editor, select Tools > References, and check the check box titled vbabork2 1.0 Type Library.

If this option is not displayed, ensure that vbabork2.dll exists in the SecureLogin directory. If it does not, re-install SecureLogin, making sure to select Terminal Launcher.

If the vbabork2 1.0 Type Library option displays, you must register it.

- 1** Open a DOS shell.
- 2** Enter **regsvr32** followed by the path to vbabork2.dll.

For example, enter

```
regsvr32 "C:\Program Files\SecureLogin\ vbabork2.dll"
```

A message should indicate that DllRegisterServer succeeded.

- 3 Add the reference to this module in Reflection as described above.

Only one session can be launched at a time using SecureLogin. To run the script, you must run the SecureLogin script macro after the session has been opened. This can be done automatically in Reflection by selecting Connect Macro from the Connection Setup menu.

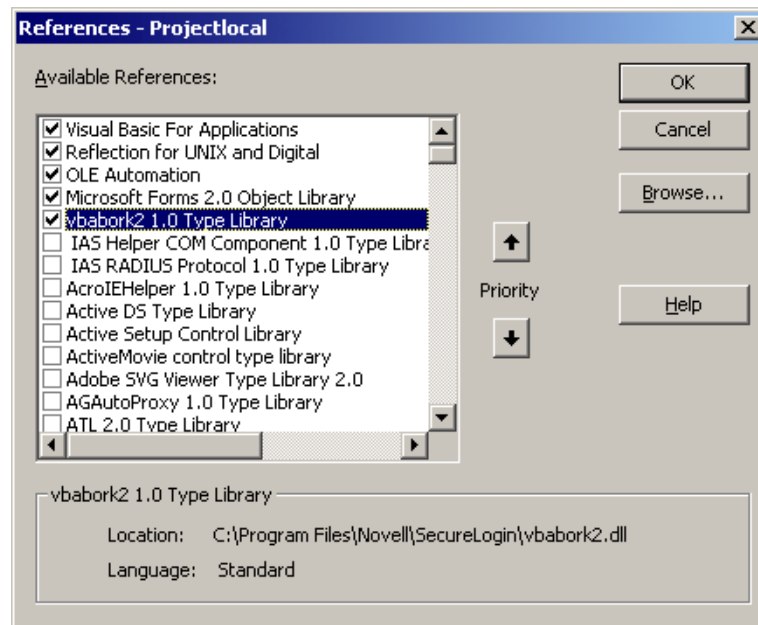
By doing this procedure, the SecureLogin script macro will run every time the session is opened. Without this procedure, you will need to manually run the script macro when you want to run the script.

To run the script, you must set up the emulator in Terminal Launcher.

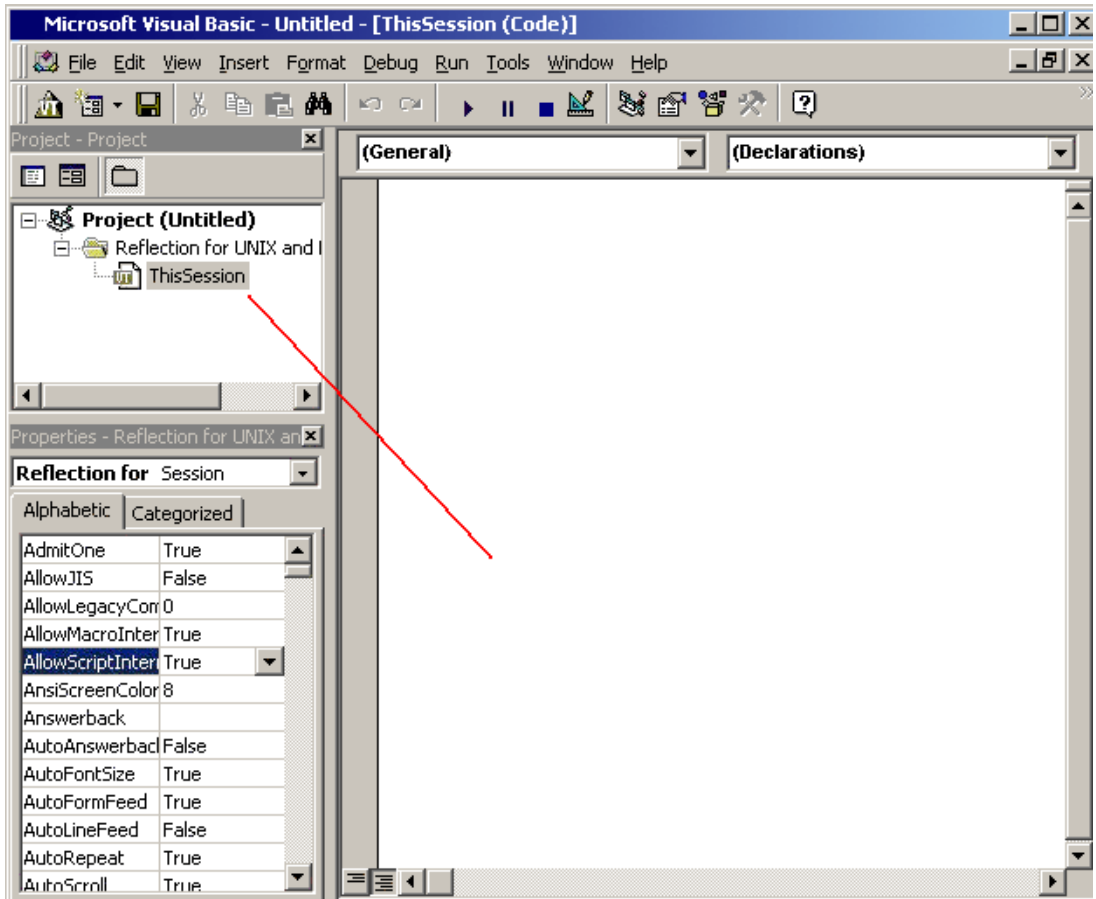
- 1 Launch Terminal Launcher, then click Edit Available Emulators. Set up the emulators as usual, but set the HLLAPI type to None.
- 2 Select the emulator.
- 3 Type anything in the hllapi dll and HLLAPI Function boxes.
- 4 Click OK, then click Done.

Configuring Reflection 9 for UNIX and Digital

- 1 Open Reflection for UNIX and Digital.
- 2 Click Macro at the top of the screen, then select Visual Basic Editor.
- 3 After the editor loads, click Tools at the top of the screen, then click References.
- 4 Scroll to vbabork2 1.0 Type Library, then check the check box.



- 5 Return to the main editor screen, then double-click This Session, which is on the far left of the screen.



6 Copy and paste the macro into the editor.

You can type text in the right pane of the screen. Type or copy and paste the following macro:

```

Sub SecureLogin()
    Dim SecureLoginObject As ISLBroker
    Dim ReturnCode As Long
    Dim Data As String
    Dim targ As Long
    Dim FunctType As Long
    Dim CR As String
    Dim temp As String

    Session.Wait 0.1 'The waits are necessary for the screen
    to be updated.

    Set SecureLoginObject = New SLBroker
    CR = Chr$(rcCR) ' Chr$(rcCR) = Chr$(13) = Control-M
    SecureLoginObject.LoadScript
    While (1 = 1)
        FunctType = 0
        'retrieve command from VBABork
        SecureLoginObject.GetCommand FunctType, targ, Data
        If FunctType = SecureLoginObject.SetCursor Then
            ' SetCursor is not supported
            ReturnCode = 0
        ElseIf FunctType = SecureLoginObject.TypeText Then
            If (StrComp(Data, "@E", vbTextCompare) = 0) Then
                Session.Transmit CR
            End If
        End If
    End While
End Sub

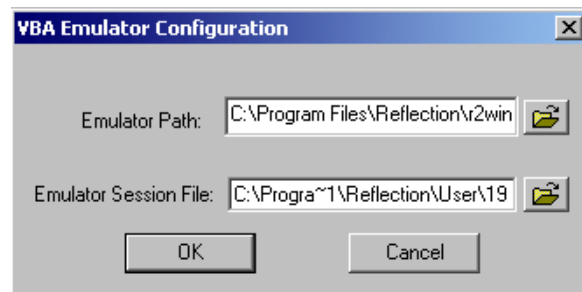
```

```

Else
    Session.Transmit Data
End If
Session.Wait 0.1
ReturnCode = 0
ElseIf FunctType = SecureLoginObject.ScanForText Then
    bResult = Session.FindText(Data,
    Session.ScreenTopRow, 0)
    ReturnCode = 0
    If bResult = True Then
        ReturnCode = 1
    End If
Else
    ' End of script
    GoTo ErrorHandler
End If
Session.Wait 0.1
SecureLoginObject.SetReturnCode ReturnCode
Wend
ErrorHandler:
End Sub

```

- 7** Click File > Close, then click Exit to exit from the editor.
- 8** Click Open Connection at the top of the screen, then click Connection Setup.
This is where you set up the host name and protocol that you are connecting to the mainframe.
- 9** Click Connect Macro and browse to the macro you just created.
- 10** Open Tlaunch.exe, then click Edit Available Emulators > New.
This step creates a new emulator.
- 11** Select a VBA emulator type, then type the required settings.



- 12** Create a script for the emulator to use.

9

Using Password Policies

SecureLogin enables you to create a password policy and have SecureLogin enforce it. The policy ensures that the values of the variables comply to certain rules governing their composition. Although this feature is called “password policies,” these policies may be used on any variables, not just password variables.

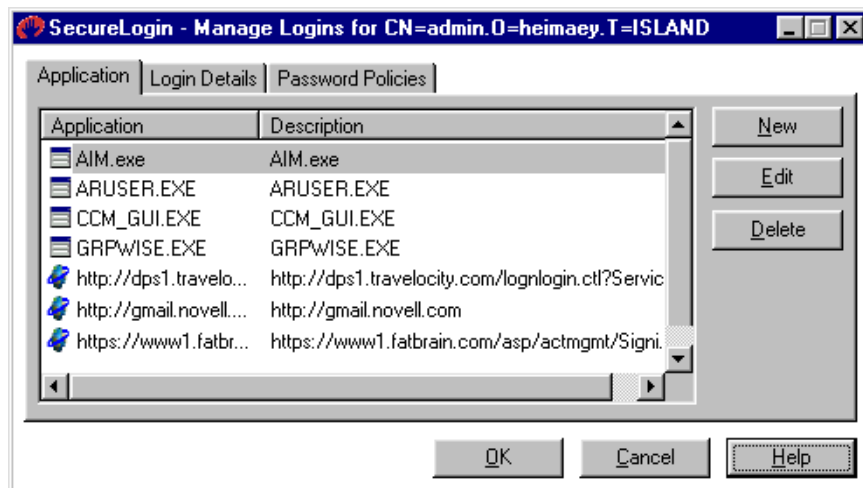
This section contains information about the following:

- ♦ [“Creating a Password Policy” on page 145](#)
- ♦ [“Restricting Variables in Policy Scripts” on page 148](#)

Creating a Password Policy

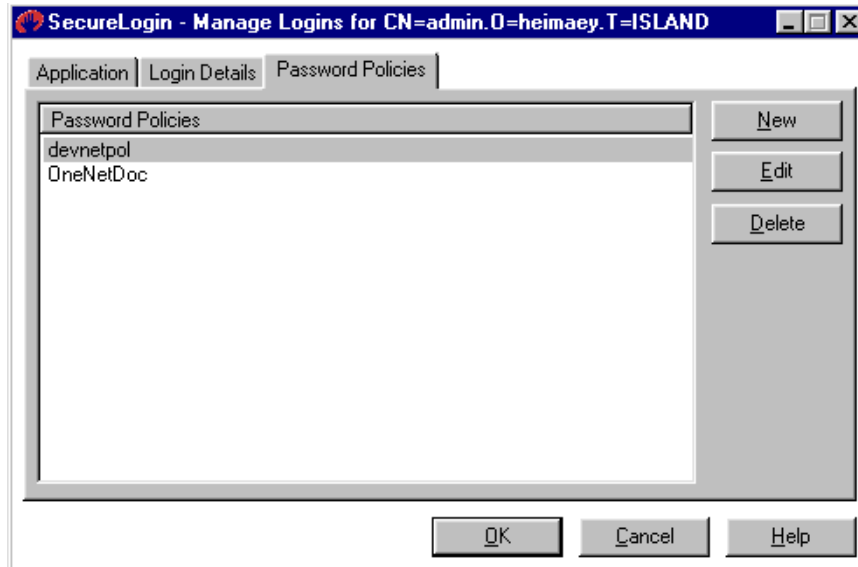
To create a password policy script, create a new script by using the user administration tool or ConsoleOne™. The following procedure is for the user administration tool.

- 1 Right-click the SecureLogin icon on the system tray and then click Manage Logins.
- 2 Select the application that you will create the script for and then click Password Policies.



- 3 (Conditional) Click New.

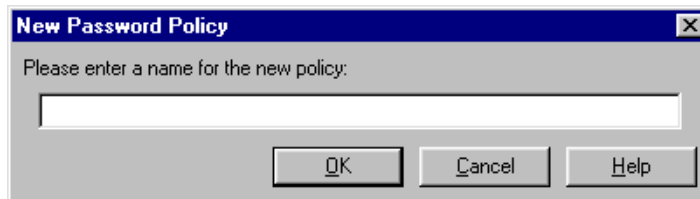
If you have already created a password policy that meets the requirements that you intend, select it.



- 4 Enter a name and then click OK.

For example, for the DevNet application, enter devnetpol instead of DevNet. Describe the application but don't use the same name used on the Login page.

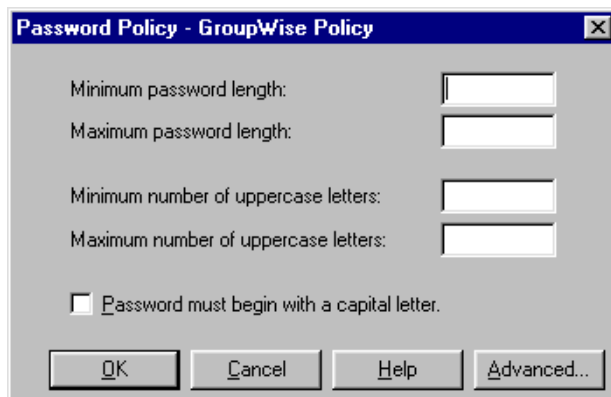
The following graphic illustrates the dialog box for a new name:



- 5 Select the new policy name, click Edit, type values, and then click OK.

The values allow you to set certain criteria or rules. For example, the password must be a certain length and it cannot contain certain characters

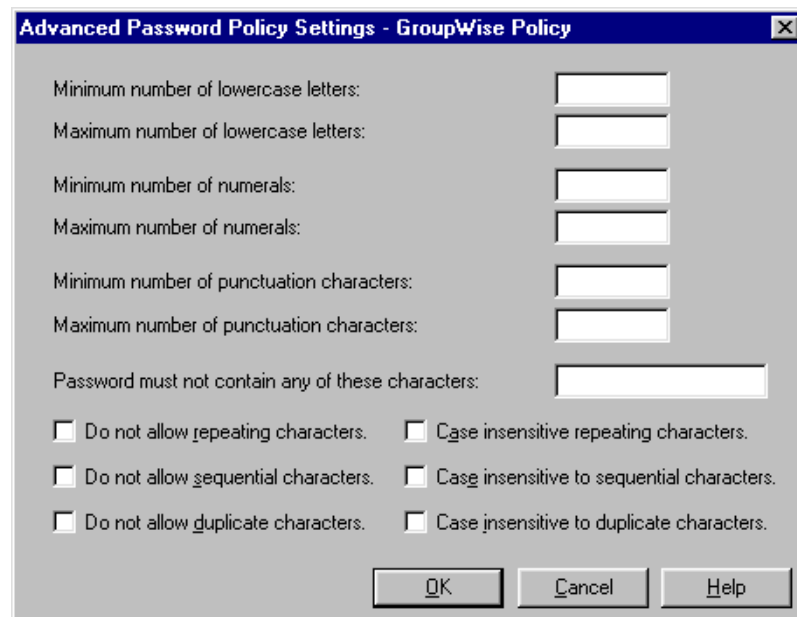
The following figure illustrates values that you can enter:



- 6 (Optional) Set advanced policies.

Click Advanced and type values.

The following figure illustrates the dialog box for advanced policies:



The following characters satisfy the punctuation setting:

Character	Character Name
~	Tilde or swung dash
!	Exclamation mark
@	"At"
#	Hash or pound
\$	Dollar
%	Percent
^	Caret
&	Ampersand
*	Asterisk
	Space
()	Parentheses
_	Underscore
+	Plus
	Delimiter or delimiter bar
-	Hyphen

Character	Character Name
=	Equals
\	Backward slash or backward diagonal
{ }	Braces or curly brackets
[]	Brackets
:	Colon
;	Semicolon
“	Quotation mark
'	Single quotation mark
<	Greater than
>	Less than
?	Question mark
/	Slash, diagonal, or slant
,	Comma
.	Period or full stop
‘	Grave or accent grave mark

Restricting Variables in Policy Scripts

To restrict a variable to a particular password policy script:

- 1** Select the script that contains the variable you want to restrict.
- 2** Add the following line at the top of the script:

```
RESTRICTVARIABLE variable_name password_policy
```

To restrict multiple variables, you just need to add multiple RestrictVariable commands.

The *variable_name* parameter can be a normal variable (for example, \$Password) or a runtime variable (for example, ?temp). This flexibility can be useful if you change a password by using a runtime variable and then set a normal variable to the value of the runtime variable.

Adding the RestrictVariable command is all you need to do to ensure that a variable will comply with the policy. The value entered will be rejected if it does not comply with the policy set for that variable, regardless of whether the variable is being added or changed through SecureLogin or through a script that is running.

If the value being changed by a user is not accepted, a message informs the user as to why the value wasn't permitted. If the value is being set through the ChangePassword command being run in automatic (random) mode, the value generated will comply with the policy.

In some cases, a policy may be created where no acceptable values exist. When this occurs, an error will be displayed when the ChangePassword command tries to generate a password.

For more information on the RestrictVariable command, see [RestrictVariable](#) in *SecureLogin Script Commands*.

Values will not be forced to comply with password policies if you use the SecureLogin SET command to set them.

Example Password Policy Scripts

Example 1

```
MAXPASSWORDLENGTH 8
MINPASSWORDLENGTH 8
MAXPUNCTUATION 0
MINPUNCTUATION 0
MAXUPPERCASE 8
MINUPPERCASE 0
MAXLOWERCASE 8
MINLOWERCASE 0
MAXNUMERALS 8
MINNUMERALS 0
```

This password policy indicates that the password must be exactly 8 characters long and contain no punctuation characters.

The password asdf4jB8 is acceptable.

The password aasdf5\$n is unacceptable because it contains a punctuation character.

Example 2

```
MAXPASSWORDLENGTH 16
MINPASSWORDLENGTH 6
MAXPUNCTUATION 8
MINPUNCTUATION 0
MAXUPPERCASE 16
MINUPPERCASE 1
MAXLOWERCASE 16
MINLOWERCASE 0
MAXNUMERALS 16
MINNUMERALS 0
BEGINWITHUPPERCASE
DISALLOWEDCHARACTERS @&
```

This password policy indicates that the password must be between 6 and 16 characters long. It must contain at least one uppercase character. It can contain no more than 8 punctuation characters. It must begin with an uppercase character, and it may not contain either the @ character or the & character.

The password R48iv"? is acceptable.

The password R48?- is unacceptable because it is less than 6 characters long.

Example 3

```
MAXPASSWORDLENGTH 12
MINPASSWORDLENGTH 6
MAXPUNCTUATION 8
MINPUNCTUATION 0
MAXUPPERCASE 8
```

```
MINUPPERCASE 0
MAXLOWERCASE 8
MINLOWERCASE 0
MAXNUMERALS 8
MINNUMERALS 0
NODUPLICATECHARACTERS CASEINSENSITIVE
POSITIONCHARACTER NUMERAL 3,4,5
```

This password policy indicates that the password must be between 6 and 12 characters long. It can contain no more than 8 of any character type (uppercase, lowercase, numeral, or punctuation). No character may appear more than once in the password, regardless of case. A numeral must appear in at least one of positions 3, 4, or 5.

The password f54v9)_Q is acceptable.

The password f5v)_QF7 is unacceptable because it has no numeral in positions 3, 4 or 5, and the letter F occurs in positions 1 and 7.

10 Troubleshooting SecureLogin

This section contains information on frequently asked questions.

For information on error codes, see [“Error Codes for LDAP” on page 154](#) and [Appendix C, “Error Codes,” on page 173](#).

Changing the Startup Order

How can I change the startup order of applications? I placed an application in the Startup folder, but SecureLogin doesn't recognize it.

Answer: Most likely, a password protected application is starting before SecureLogin is initialized and able to process login requests. Try one of the options in [“Changing the Startup Order of Applications” on page 87](#).

Entering a Passphrase

What's a passphrase? After I set up SecureLogin, a dialog box instructs me to enter my passphrase. However, the entry fields don't let me know which box is the passphrase box.

Answer: Actually, SecureLogin has two passphrase components. The question that you enter is the passphrase question. The answer that you provide is the passphrase answer. If someone changes your password and tries to log in as you, that person must correctly answer the passphrase answer to the passphrase question that displays at relogin.

If you encounter the passphrase question, just answer it. An additional dialog box will instruct you to enter your NDS[®] or eDirectory password, so that you can log in. If appropriate, you can then reset your password so that whoever reset it can't log in as you.

Can't Log In Again to a Web Site

After changing a SecureLogin password to match a Web site password, why can't I log in to the Web site?

Answer: Internet Explorer's AutoComplete function can cause this problem. Disable AutoComplete.

Scenario. While in disconnected mode, Rie successfully enters a SecureLogin username and password for a Web site. Using a script at the Web site, Rie changes the password and then edits the SecureLogin entry, so that the SecureLogin password matches the Web site. The single sign-on login for the Web site now fails.

Rie disables Internet Explorer's AutoComplete function and is able to log in.

Troubleshooting Scripts for Web Sites

What's the best way to log in to Web sites?

Answer: Because SecureLogin recognizes a login panel on a Web page, the easiest method to create scripts for Web sites is to use the popup wizard. The second option is to run the wizard manually.

If for some reason you need to examine or modify scripts, you can use the following scripts to enable most HTML Web sites to use SecureLogin. Script One works for more than 95% of HTML Web pages.

Script One

```
Type $Username  
Type $Password password
```

The `password` flag always follows the variable that contains the password.

If the first eight letters of a variable are `password`, the password is masked. If the first eight letters of a variable are not `Password`, the entry is displayed normally, unless the Web page masks the entry with asterisks.

The following table illustrates uses of the `$password` variable:

Command	Variable	Result
Type	<code>\$password password</code>	Enters the value of the variable <code>\$password</code> and displays asterisks because the first eight letters of the variable are <code>password</code> .
Type	<code>\$juanspassword password</code>	Enters the value of the variable <code>\$juanspassword</code> , but not as asterisks, unless the Web page masks the entry with asterisks.
Type	<code>\$password4juan password</code>	Enters the value of the variable <code>\$password</code> and displays asterisks because the first eight letters of the variable are <code>password</code> .

Script Two

```
Type $Username #1  
Type $Password #2  
Click #1
```

This script is also successful for Web sites. The parameter `#1` instructs SecureLogin to enter the value of the variable `$password` into the first (from top to bottom) entry field on the page.

HINT: If a Web page uses frames, "top to bottom" might not be obvious. In this case, try different numbers until one works.

The parameter `#2` instructs SecureLogin to enter the value of the variable `$password` into the second entry field on the page.

Using the `#1` parameter with the click command instructs SecureLogin which button on the page to click.

The script submits automatically. If a problem occurs, use the following commands:

- ◆ Type \n
This option presses Enter.
- ◆ Type \n *control position* (for example, Type \n #1)
This option presses Enter for the specified button for field number. You can also try changing the #1 to #2, #3, and so on to make sure that SecureLogin presses the correct button.
- ◆ Click *control position* (for example, Click #2)
- ◆ “Submit”
This option forces a submit.

For more information, see **Click** in *SecureLogin Script Commands*.

Logging in to Difficult Web Sites

Why is SecureLogin unable to log me in to some Web sites?

Answer: You might need to create a registry entry value.

- 1** Create a DWORD value in the registry called IELoggingMode.

Create this command under HKEY_CURRENT_USER\Software\Protocom\SecureLogin.

- 2** Set the DWORD value to 1.

When this value is turned on, no scripts will be run. However, when you go to any Web page, this setting logs the messages that it receives. The log file gets saved as c:\IEPageTest.txt. Information in the log file can help with difficult sites.

System Tray Icon Stays Active

Why does the SecureLogin icon remain active? I used the User Preferences page to turn off the SecureLogin icon on the system tray. Then I refreshed the data.

Answer: This setting is only read at startup. After you restart your workstation, the system tray won't display the icon.

No Attribute Mapping Tab

Why can't I locate the attribute mapping tab on the property page for the LDAP Group object? I'm trying to map Protocom-SSO-entries to the existing Prot.SSO entry.

Answer: You probably haven't installed the LDAP snap-in to ConsoleOne™. Download the snap-in from [Product Downloads \(http://download.novell.com/filedist/PublicSearch\)](http://download.novell.com/filedist/PublicSearch).

- 1** In the Search for a Download section, search using Category.
- 2** From the Choose a Category drop-down list, select ConsoleOne Snap-ins.
- 3** From the Choose a Platform drop-down list, select NetWare.
- 4** Click Submit Search.
- 5** Under the eDirectory section, click Download.

Error Codes for LDAP

Where can I find information about error codes for LDAP?

Answer: Get information from [LDAP and NDS Integration \(http://developer.novell.com/ndk/doc_jldapunx.htm\)](http://developer.novell.com/ndk/doc_jldapunx.htm), in the Novell Developer Kit (NDK). Navigate to an option under LDAP Server Return Codes:

- ◆ LDAP Client Return Codes
- ◆ LDAP Server Return Codes
- ◆ LDAP Result Code Structure

Resolving Error -602

How do I fix error -602?

Answer: Error -602 is “No Such Value”. This is an NDS error code. Search on 602 (no -) at [Novell Support Knowledgebase \(http://support.novell.com/search/kb_index.jsp\)](http://support.novell.com/search/kb_index.jsp).

Resolving Error -672

Why did I receive error -672? When I logged in for the first time, I entered a passphrase and answer, but I couldn’t save the data to eDirectory™.

Answer: -672 is an NDS error: Access Denied. Most likely, Novell® SecureLogin tried to write the passphrase information to the prot: * attributes but the user didn’t have sufficient rights. The administrator needs to run the rights assignment part of schema.exe, which is typically located in the c:\Program Files\novell\securelogin directory.

Resolving Error -1644

Why do I get error -1644 during installation?

Answer: You are probably installing on a Windows* 2000 workstation. To install the SecureLogin client there, you must have Power User or Administrator privileges to the workstation.

Error Parsing Line

How do I resolve “Error parsing line”?

Answer: Put a MessageBox command between lines of the script.

If a script breaks down, SecureLogin typically displays “Error parsing line” to inform you that the script isn’t working. However, occasionally the script breaks down even though there is no error parsing a line. By putting the MessageBox command between lines of the script, you can see exactly where the script stops functioning.

The following sample illustrates using the MessageBox command.

```
Type $username
MessageBox "This is the first message box after username"
Type $password
MessageBox "after password"
```

```
Click #1  
Messagebox "after click#1"  
Click #2  
Message box "after click#2"
```

If the message box with the text “This is the first message box after username” appears, you know that the first line of the script executed successfully. To allow the script to continue to the next line, click OK on the message box.

For more information, see [MessageBox](#) in *SecureLogin Script Commands*.

Program Conflict

What causes the Program Conflict message? During installation, I checked Start SecureLogin Now. However, I get this message: “Unable to load all entry points from access library (ssman.dll). Please check that it is in the path and the correct version.”

Answer: If the Program Conflict message appears during installation, Make sure that previous Novell Single Sign-on software components have been uninstalled or otherwise removed from the system.

Also, delete the following entries (if they exist) from the registry:

- ◆ HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Single Sign-on
- ◆ HKEY_LOCAL_MACHINE\SOFTWARE\Passlogix

A

Finding Control IDs and Offsets

This section provides information on the following:

- ◆ “Finding Input and Output IDs” on page 157
- ◆ “Finding Offsets” on page 165

Finding Input and Output IDs

Every option (for example, File and Edit) in an emulator’s menu has a unique Control ID number. Terminal Launcher uses these IDs to simulate the selection of options.

Terminal Launcher uses

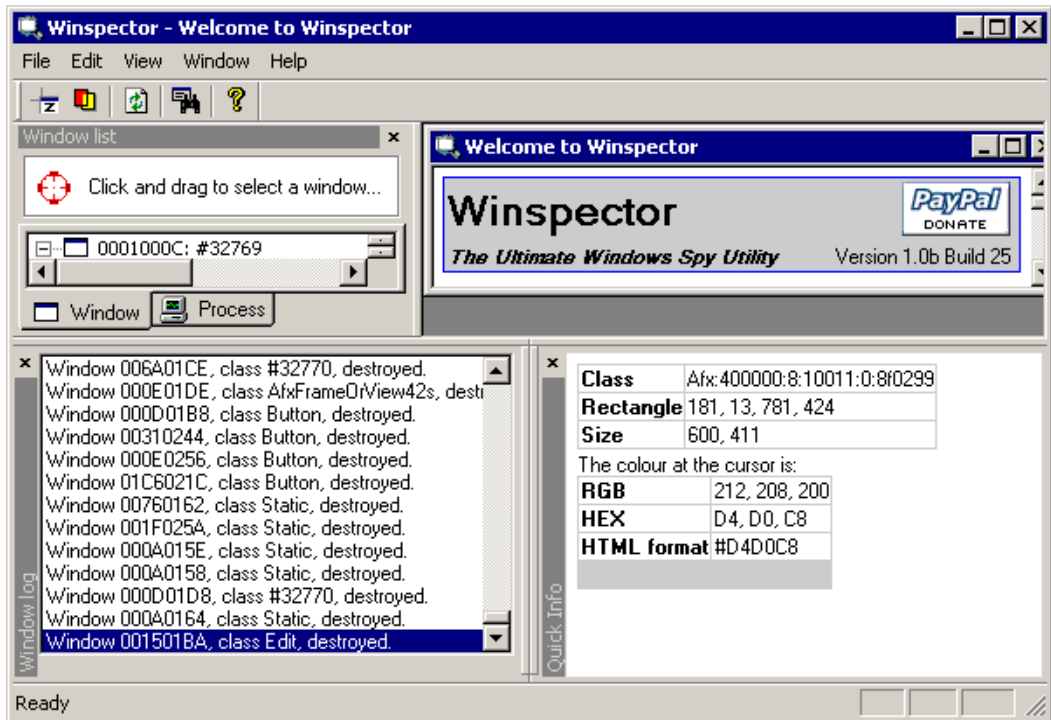
- ◆ The IDs of the Select All and Copy functions as the Output IDs
- ◆ The ID of the Paste function as the Input ID

To find these numbers, use Winspector, available at the [Gipsysoft Web site \(http://www.gipsysoft.com/articles/winspector/\)](http://www.gipsysoft.com/articles/winspector/), or Spy++, available from Microsoft’s developer Web site.

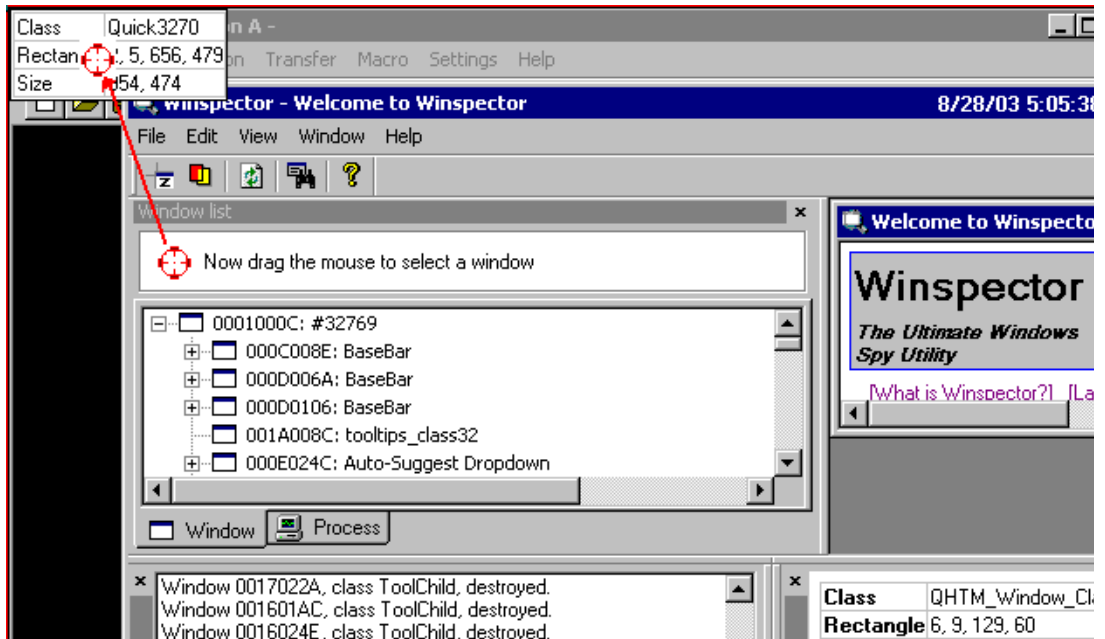
This section explains how to use Winspector.

Setting Up Winspector

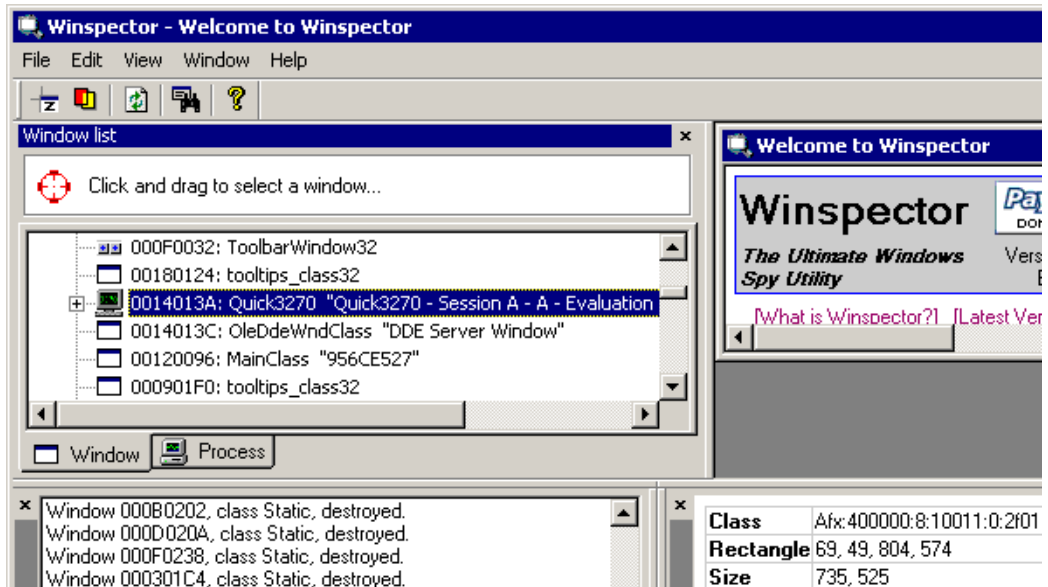
- 1** Run the emulator. Don’t close it.
In this example, the emulator is Quick32.
- 2** Run Winspector.



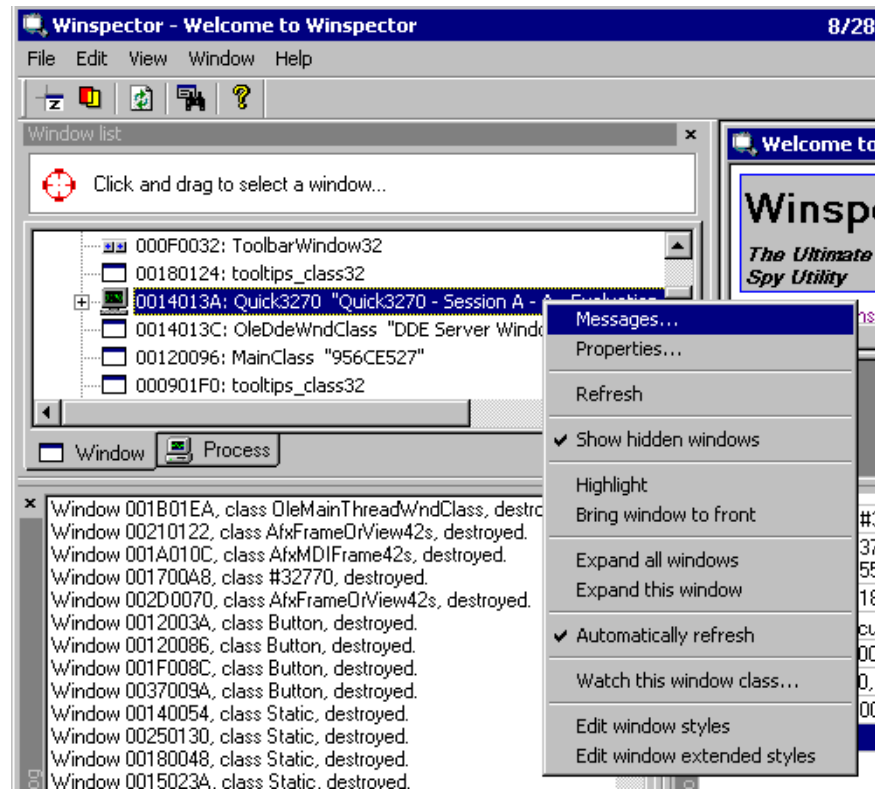
- 3 Arrange the windows so that emulator window and Winspector window are both visible on the screen.
- 4 Click and drag the Winspector icon to the title bar of the emulator, then release the mouse.



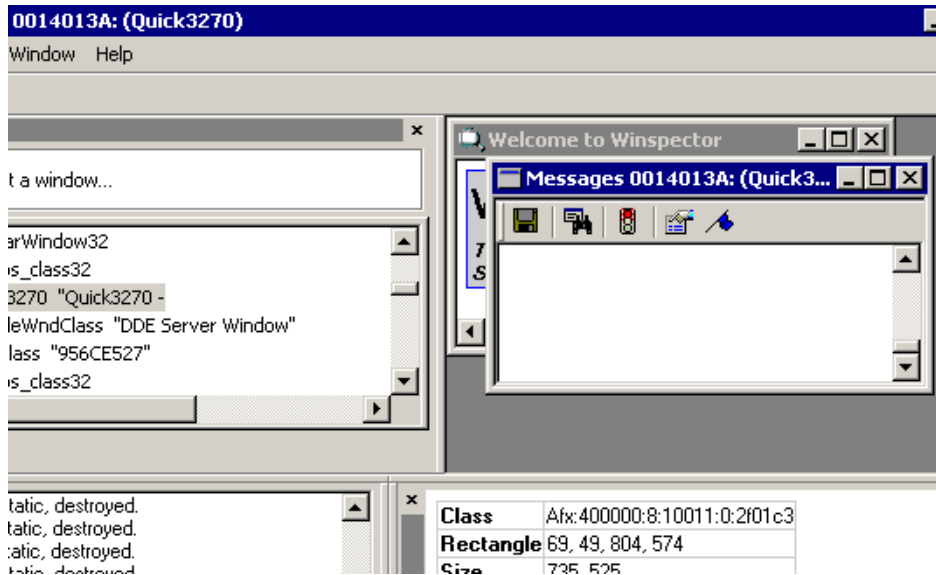
- 5 Verify that the title displayed on the emulator's title bar is listed among applications in Winspector's Click and Drag To Select a Window pane.



6 Right-click the emulator name, then click Messages.

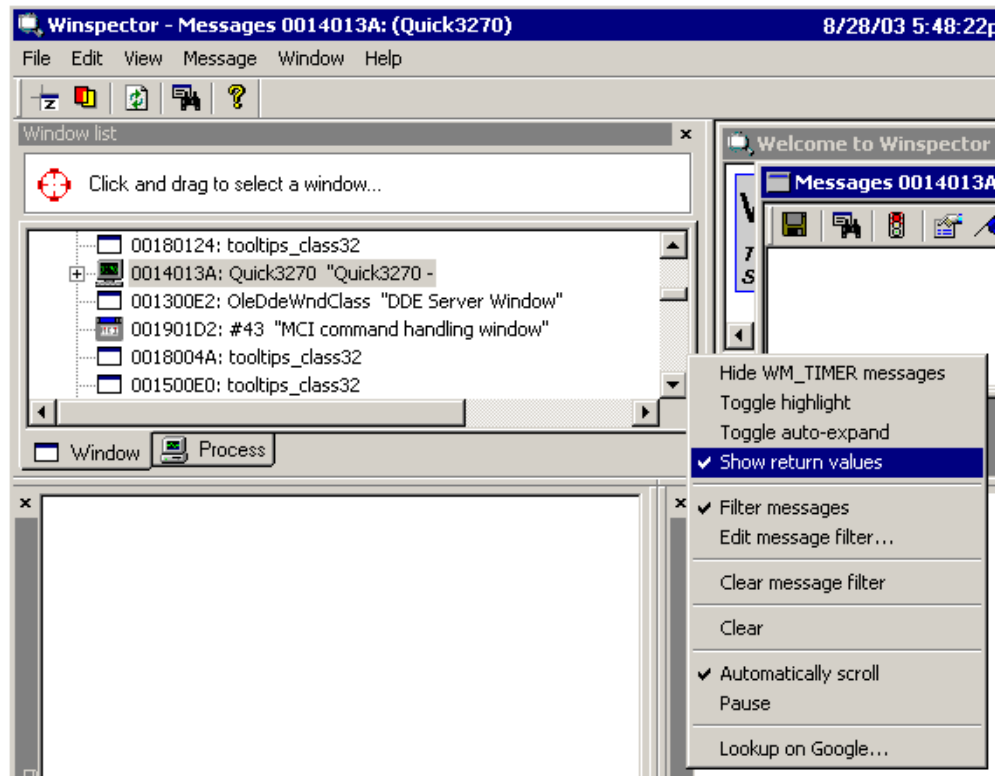


With the Messages window open, Winspector is ready to capture and log messages.

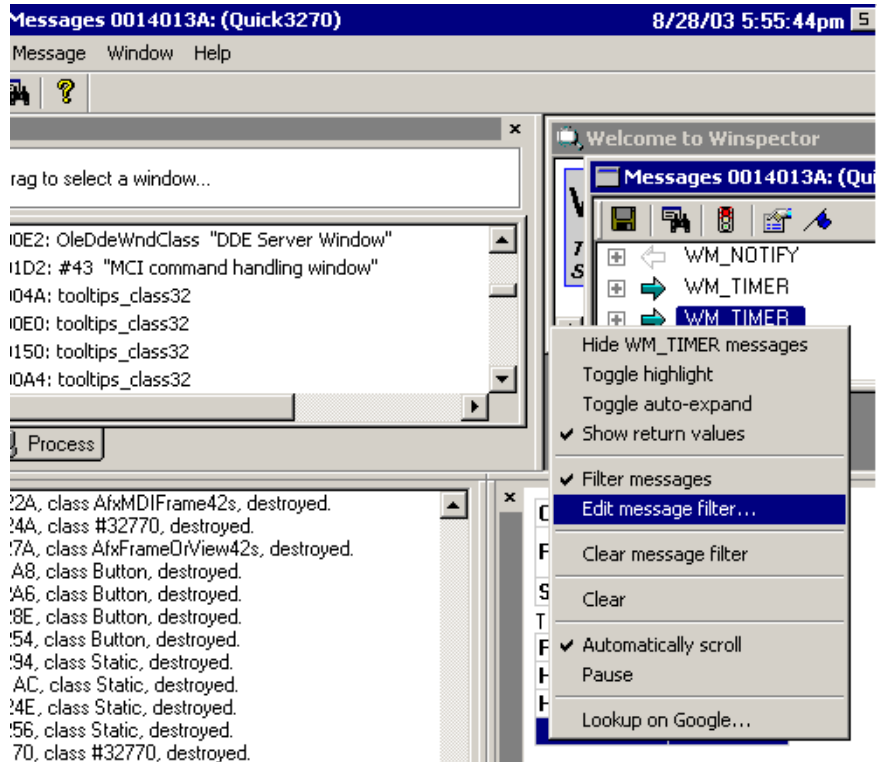


7 Limit what Winspector captures and logs.

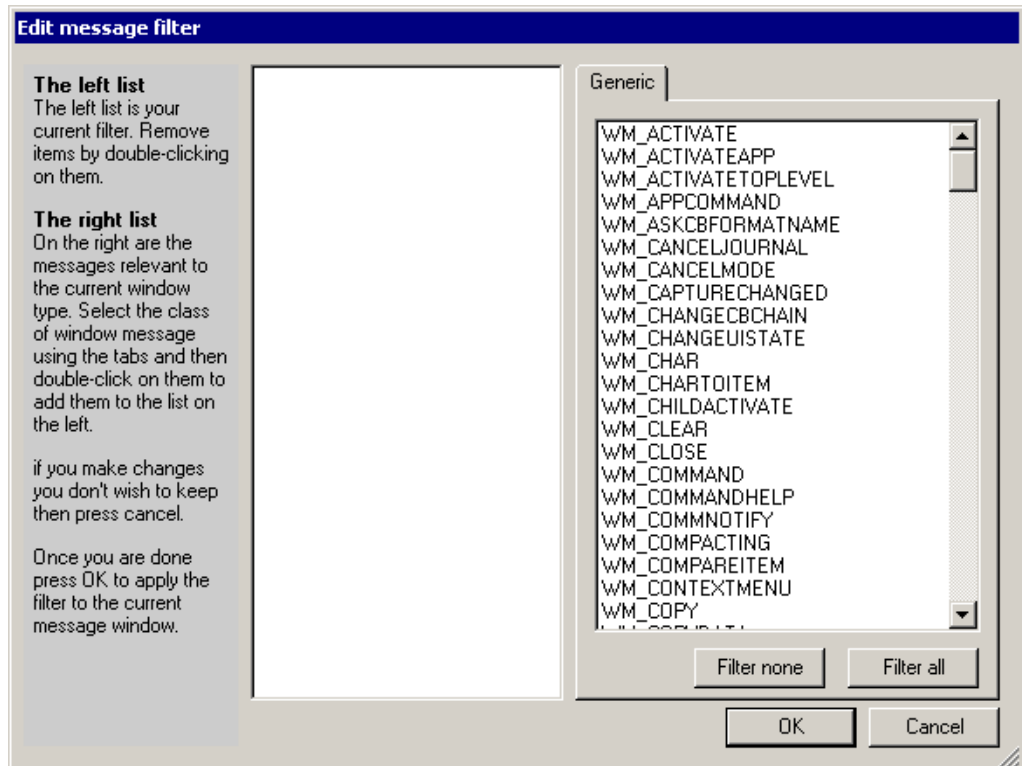
7a Right-click in the Messages window, then uncheck Show Return Values.



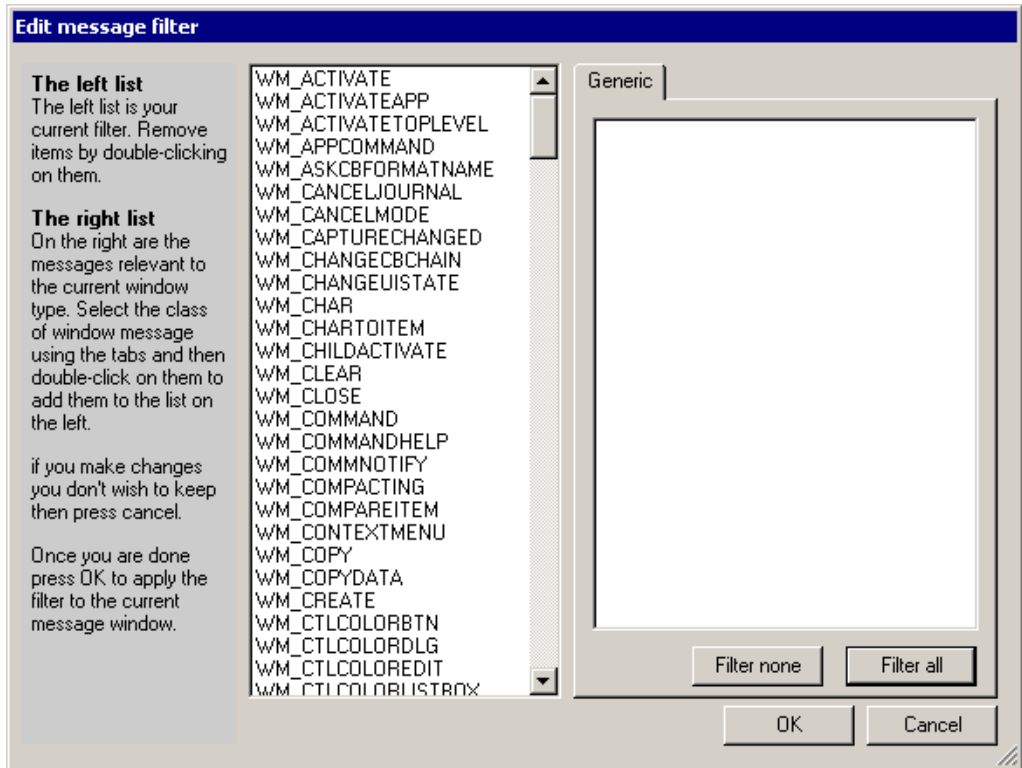
7b Right-click in the Messages window, then click Edit Message Filter.



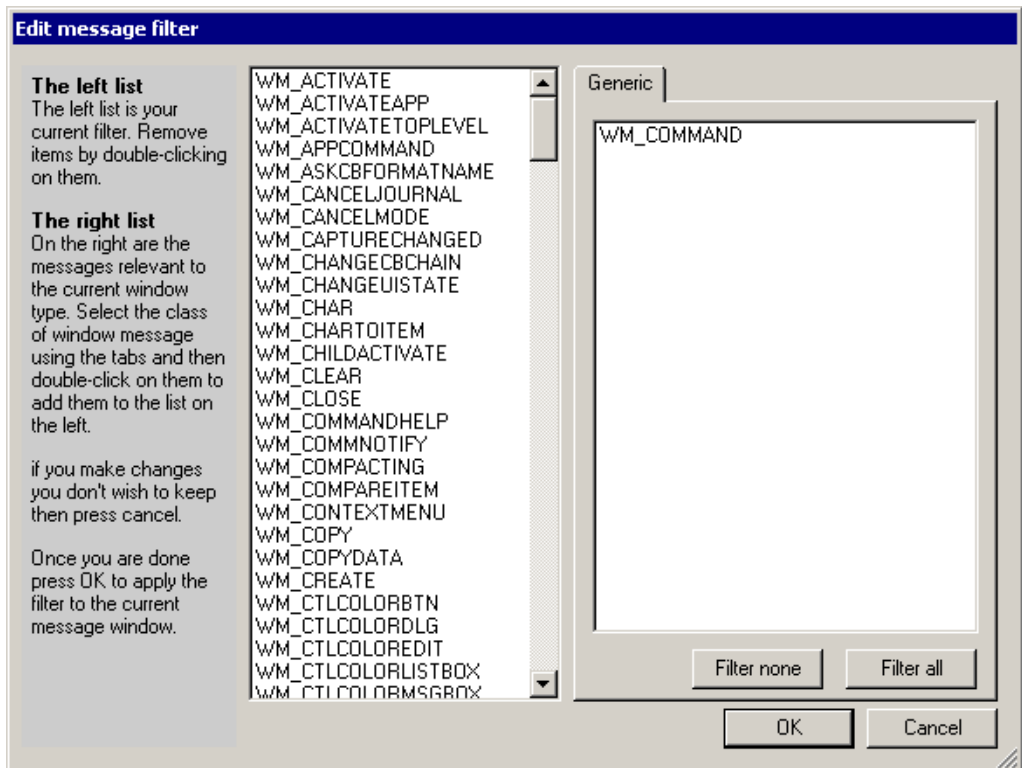
7c From the Edit Message Filter Window, click Filter All.



When all messages are filtered, WinInspector rejects all messages. The Messages window won't be cluttered with unwanted messages.

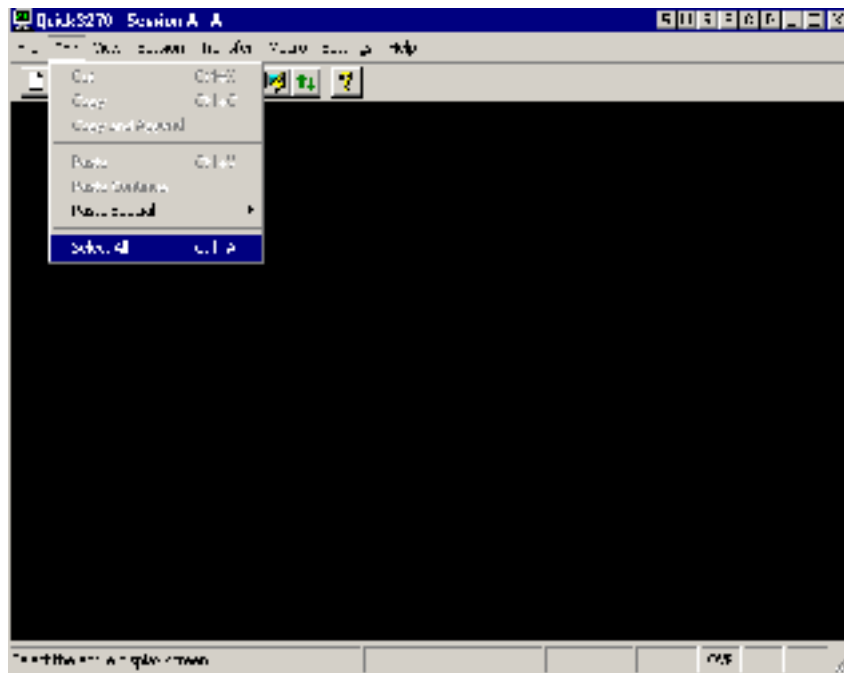


7d Scroll to and double-click WM_COMMAND, then click OK.

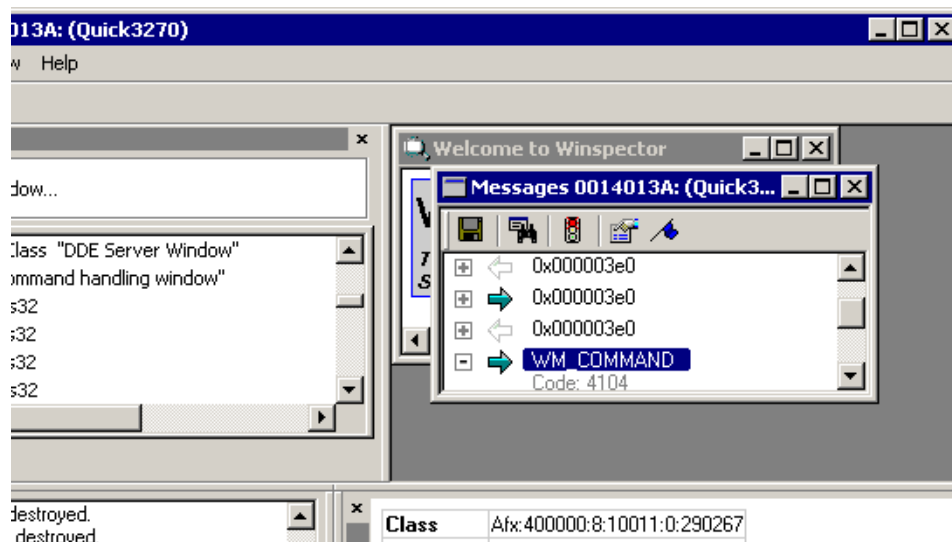


Viewing Control ID Numbers

- 1 (Conditional) From the emulator's Edit drop-down list, click Select All.
If the emulator doesn't have a Select All option, skip this step.



- 2 In Winspector's Messages window, expand WM_COMMAND.



- 3 View the Control ID.
Winspector uses the Code line to display the control ID. In this example, the control ID for Select All is 4104.
- 4 Find the Control IDs for the Copy and Paste functions by repeating Step 1 through Step 3.
- 5 Configure Terminal Launcher by adding the Control IDs.

Using Alternatives to Control ID Numbers

In the following situations, use an alternative to Control ID numbers:

- ◆ An emulator doesn't have standard Control IDs. Either no IDs are present or all functions have the same ID.
- ◆ The Control IDs in the Terminal Launcher configuration don't work.
- ◆ You don't have access to software such as Winspector or Spy++ to find the Control IDs.

The alternatives simulate keyboard shortcuts that achieve the same outcome. The keystrokes that you simulate depend on the keyboard shortcuts. The following tables lists the typical shortcuts:

Function	Keyboard Shortcut
Copy	Ctrl+C
Paste	Ctrl+V
Select All	Ctrl+A

For example, if the Select All Control ID 64 and the Copy Control ID 50 don't work with Terminal Launcher, try any of the following alternatives in the Output IDs text box.

Alternative One

```
\Alt+E,S,\Alt+E,C
```

This output simulates the following:

- ◆ Alt+E (displays the Edit menu list)
- ◆ S (selects Select All)
- ◆ Alt+E (brings up the Edit menu list)
- ◆ C (selects Copy)

Keys are not case sensitive.

Alternative Two

```
\Ctrl+a,\Ctrl+c
```

This output simulates the following:

- ◆ Control+A (a keyboard shortcut for Select All)
- ◆ Control+C (a keyboard shortcut for Copy)

Alternative Three

```
\Alt+e,\|40,\|40,\N,\Alt+e,\N
```

This output simulates the following:

- ◆ Alt+E (displays the Edit menu list)
- ◆ Down-arrow
- ◆ Down-arrow (scrolls down to Select All)

- ◆ Enter (selects Select All)
- ◆ Alt+E (displays the Edit menu list)
- ◆ Enter (selects Copy)

All of these alternatives might or might not work with particular emulators. They will all need customizing to suit the emulator. You can adapt the alternatives to suit any requirement (for example, selecting other menu items).

Finding a Terminal Launcher configuration, including finding alternatives to the Control IDs, requires trial and error, which becomes easier with experience.

When you use alternatives, the increased number of steps slows the Terminal Launcher process slightly.

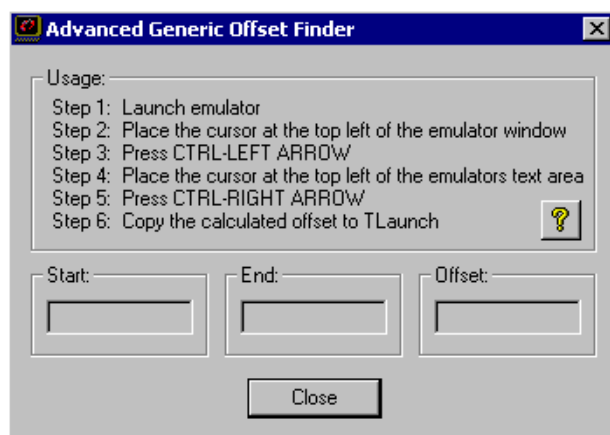
Finding Offsets

Advanced Generic emulators don't have a Select All function. To compensate, you must gather information by dragging-and-dropping across the emulator's screen, from the top-left corner to the bottom-right corner of the display area. Therefore, you need to know the offsets, so that you begin dragging at the correct location. Otherwise, the drag-and-drop process might cause unexpected behavior (Drag your tool bar or window off the screen, close an application, resize your desktop...).

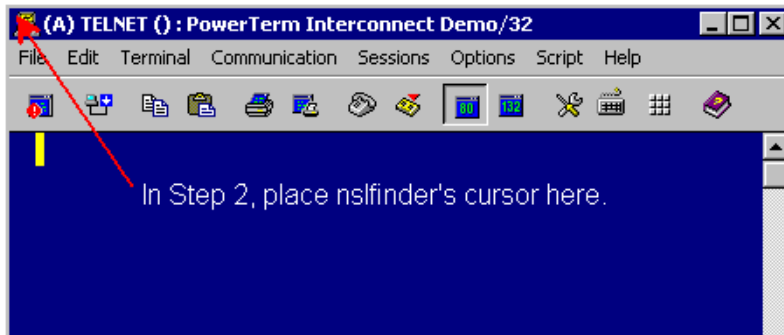
You can easily find offsets required for Advanced Generic emulators by using `nslfinder.exe`. Download this utility from [Novell's Support Web site \(http://support.novell.com/servlet/filefinder?name=nslfinder.exe\)](http://support.novell.com/servlet/filefinder?name=nslfinder.exe).

To find offsets:

- 1 Open the emulator, then open `nslfinder.exe`.

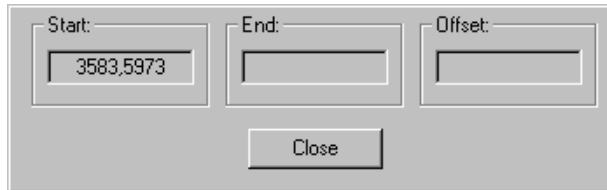


- 2 Place `nslfinder`'s cursor in the top left corner of the emulator's window.

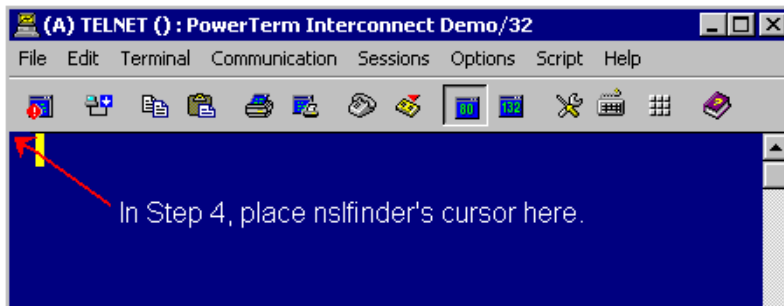


- 3 Type Ctrl+Left-arrow.

Nsfinder pastes the value into the Start text box.

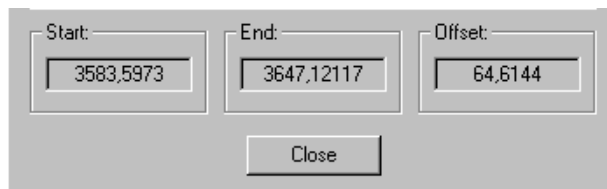


- 4 Place nsfinder's cursor in the upper left corner of the emulator's text pane.

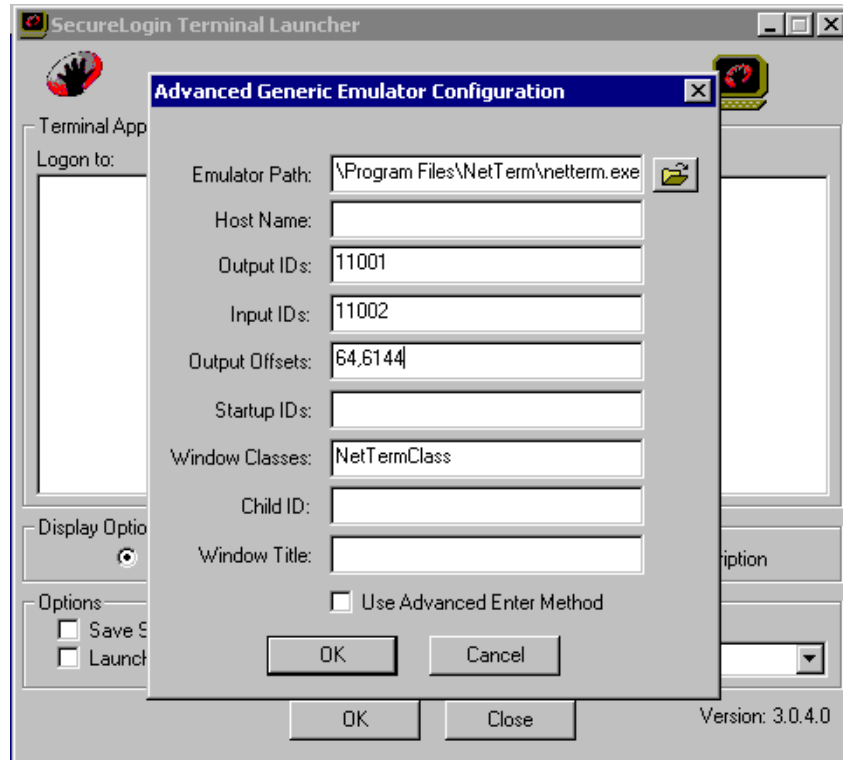


- 5 Type Ctrl+Right-arrow.

Nsfinder pastes the end value of the emulator's tool bar and menu areas into the End text box and pastes the offset numbers into the Offset text box.



- 6 Copy the offset values into Terminal Launcher's Output Offsets text box.



B

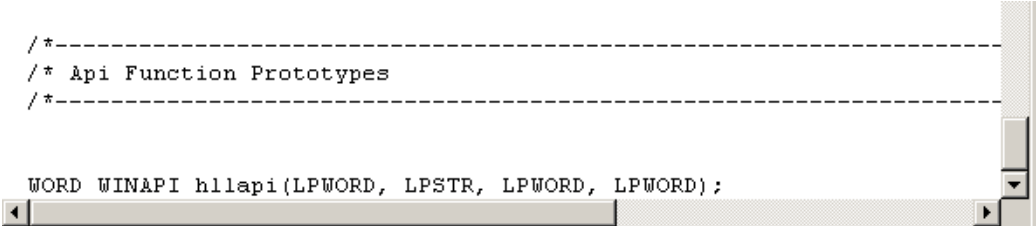
Finding HLLAPI Types

This section provides information on the following:

- ♦ “Using Header Files” on page 169
- ♦ “Using Dependency Walker” on page 169

Using Header Files

If the emulator has an accompanying .h file, search it to find the HLLAPI function. The function might be named hllapi, HLLAPI, or some variation. The following figure illustrates the hllapi notation in the .h file for Quick3270:



```
/*-----  
/* Api Function Prototypes  
/*-----  
  
WORD WINAPI hllapi(LPWORD, LPSTR, LPWORD, LPWORD);
```

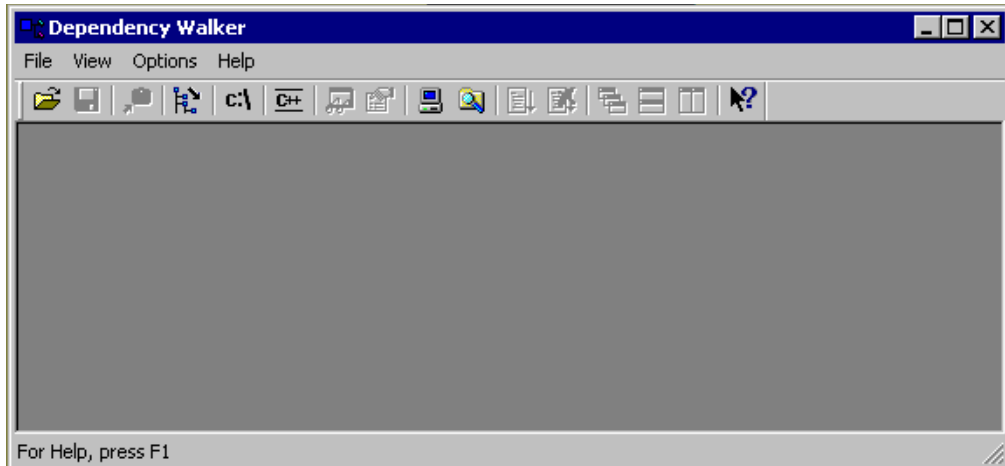
Using Dependency Walker

Dependency Walker (depends.exe) can help you configure terminal emulators for SecureLogin’s single sign-on functionality.

NOTE: Dependency Walker doesn’t work with 16-bit HLLAPI emulators.

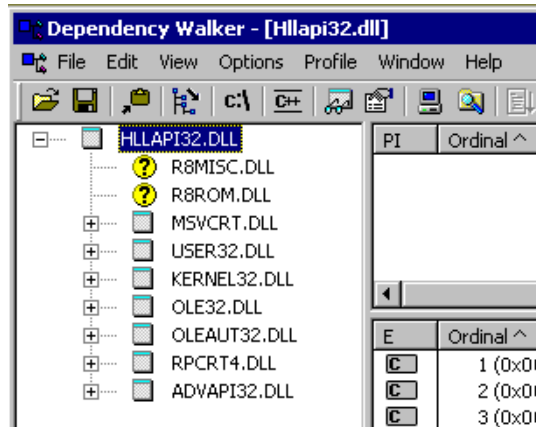
Dependency Walker is available from the [Dependency Walker Web site \(http://www.dependencywalker.com\)](http://www.dependencywalker.com).

- 1 Run depends.exe.



- 2 Click File > Open, then navigate to and open the hllapi.dll file.

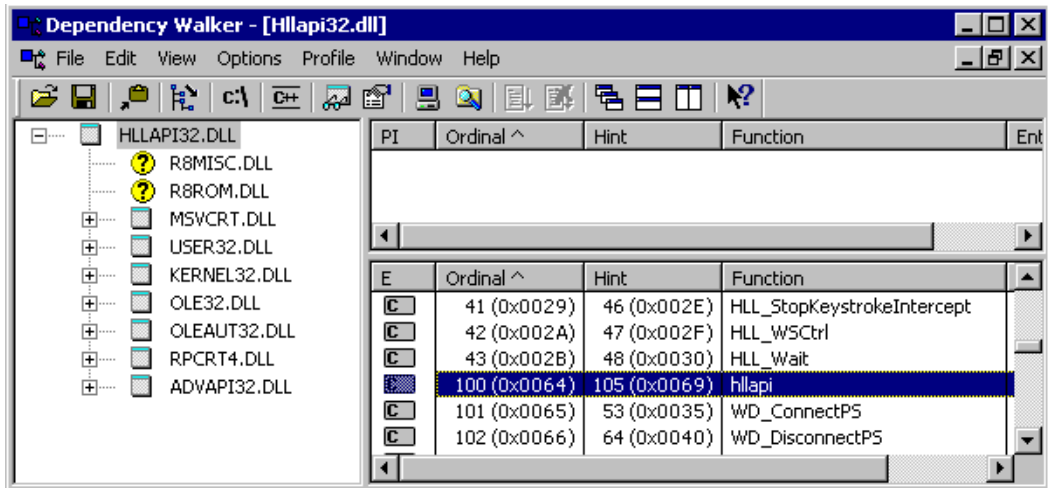
Look for a .dll file with the name "HLLAPI" (or some variation) in it. For example, the file for Quick3270 is QHLLAPI.DLL, found in the c:\Program Files\Quick3270 directory.



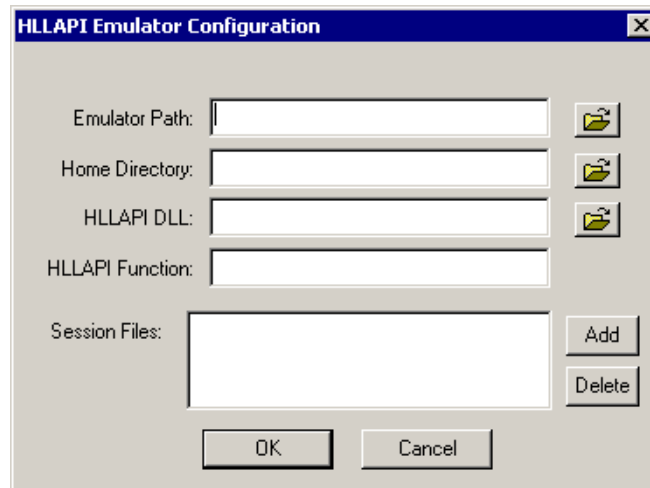
- 3 In the function exports window, scroll through the exports until you locate "hllapi".

The hllapi.dll file contains many HLLAPI functions. Use Dependency Walker to identify the HLLAPI initialization function or entry point. Typically, this function is called hllapi, Hllapi, Winhllapi, or WinHllapi. Look for some variation of hllapi.

As the following figure illustrates, the export function displayed for Quick3270 is hllapi.



4 Enter this function type in Terminal Launcher's HLLAPI Function text box.



The function is case sensitive. For example, if the function is Hllapi and you enter hllapi, the emulator won't work.

C

Error Codes

This section contains error codes for SecureLogin.

For help with error codes that provide possible causes along with actions you can take, see “[Error Codes with Tips](#)” on page 173.

For a list of error codes that require help from Novell Technical Services, see “[Getting Help from Novell Technical Services](#)” on page 191.

For information on error codes for LDAP, see [LDAP and NDS Integration \(http://developer.novell.com/ndk/doc_jldapunx.htm\)](http://developer.novell.com/ndk/doc_jldapunx.htm) in the Novell Developer Kit (NDK). Navigate to an option under LDAP Server Return Codes. For example, select LDAP and NDS, select LDAP Return Codes, then select LDAP Server Return Codes.

Error Codes with Tips

-102 BROKER_NO_SUCH_ENTRY

Possible Cause: You tried to load a script or variable that doesn't exist. For example, you set up Terminal Launcher to run from a shortcut or to run a particular script, but the script doesn't exist.

Action: Check that the name of the script is actually defined in SecureLogin. Verify that the name is the same as specified in the script editor.

-103 BROKER_INVALID_CLASS_CREATED

Possible Cause: You are running an earlier version. SecureLogin is trying to create a new version of the script data format that was stored in NDS.

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.

-107 BROKER_ENTRY_NOT_FOUND

Possible Cause: You tried to load a script or variable that doesn't exist.

Action: Check that the name of the script is actually defined in SecureLogin. Verify that the name is the same as specified in the script 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.

-112 BROKER_NO_SUCH_VARIABLE

Possible Cause: You are trying to use an undefined variable. Because SecureLogin isn't prompting you for the variable, data has become unusable, or some other situation is preventing the software from working as expected.

Action: Call Novell Technical Services.

-114 BROKER_PRIMARY_NOT_AVAILABLE

Possible Cause: You are not logged in to the Directory. You are using the offline cache. Therefore, some Directory functions can't be performed. (For example, you can't change your passphrase.)

Action: Log in to the Directory.

-116 BROKER_HEADER_DATA_CORRUPT

Possible Cause: Data isn't available. You might have had 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.

Action: Call Novell Technical Services.

-123 BROKER_CACHE_PASSWORD_INCORRECT

Possible Cause: You have tried to log in 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. However, if you have installed the NMAS module, the passphrase can be the passphrase answer or your current NDS or eDirectory password.

Action: Enter the correct passphrase answer or Directory password.

-129 BROKER_ENTRY_LIST_NOT_NULL

Possible Cause: The software is not working as intended.

Action: Delete the local cache file and try again.

Action: Call Novell Technical Services.

-131 BROKER_SYM_LIST_NOT_NULL

Possible Cause: Memory is not being handled as expected.

Action: Call Novell Technical Services.

-138 BROKER_SYMBOL_DATA_CORRUPT

Possible Cause: SecureLogin is unable to use the data in the local cache file or in the Directory.

Action: Delete the local cache file and try again.

Action: Call Novell Technical Services.

-140 BROKER_SCRIPT_DATA_CORRUPT

Possible Cause: SecureLogin is unable to use data in scripts.

Action: Delete the local cache file and try again.

Action: Call Novell Technical Services.

-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 that 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 haven't 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 haven't 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 has rights to the cache directory.

-174 BROKER_CACHE_SAVE_FAILED

Possible Cause: SecureLogin tried to save data to the offline cache but is unable to.

Action: Assign rights so that the specified user has rights to the cache directory.

-175 BROKER_CACHE_SECRETS_INCORRECT

Possible Cause: The offline cache password is incorrect. The key that is being used to decrypt the cache file is not the key that the cache file was encrypted with.

Possible Cause: After logging in as a user and creating a cache file, you went to another workstation, reset your passphrase, and logged in. You receive this error when you return to the original workstation.

Action: Delete the cache file.

-176 BROKER_PUBLIC_KEY_READ_FAILED

Possible Cause: SecureLogin is unable to read the public key from NDS or eDirectory.

Action: Troubleshoot NDS or eDirectory.

-179 BROKER_RTVALUE_DOES_NOT_EXIST

Possible Cause: You tried to read a runtime variable that has not been defined.

Action: Check the script. Make sure that the variable has been 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 can't read the variable.

Action: Make sure that you have spelled the attribute name correctly.

Action: Troubleshoot NDS or eDirectory.

-181 BROKER_WRONG_PASS_PHRASE

Possible Cause: You entered the wrong passphrase.

Possible Cause: You tried to change your passphrase but typed it incorrectly.

Possible Cause: You enabled a preference relating to SecureLogin's hand icon. You double-clicked the hand icon but typed the passphrase incorrectly.

Action: Enter the passphrase correctly.

-190 BROKER_NO_AUTH_DATA_FOUND

Possible Cause: Although the Prot:SSO Entry attribute has data, the Prot:SSO Auth attribute was blank. Someone deleted the Prot:SSO Auth attribute.

Action: Delete the Prot:SSO Entry attribute. SecureLogin creates these attributes the next time that you run SecureLogin.

-192 BROKER_UNABLE_TO_INSTANTIATE

Possible Cause: A module (for example, WinSSO) has tried to connect to the Combroker but is unable to.

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.

-199 BROKER_ERROR_COMMAND_NOT_HANDLED

Possible Cause: A script parser encountered an unrecognizable command.

Action: Make sure that you have spelled the command correctly.

Action: Make sure that the If/EndIf blocks match.

-201 BROKER_UNEXPECTED_END_OF_SCRIPT

Possible Cause: If/EndIf or Repeat/EndRepeat blocks don't match. SecureLogin reached the end of the script without finding an expected EndIf or EndRepeat command.

Action: Check the script. Make sure that If/EndIf and Repeat/EndRepeat blocks match.

-211 BROKER_ENTRY_ALREADY_IN_LIST

Possible Cause: You tried to add a script or variable, but a script or variable with that name already exists.

Action: Use a different name for the script or variable, or use the script editor to rename the existing script or variable.

-217 BROKER_ARG_NUM

Possible Cause: In script language, each command expects a certain number of arguments. You have used either too few or too many arguments for a given command.

Action: Check the documentation for that command. Make sure that you are passing to the command the correct number of arguments.

-219 BROKER_NOT_A_NUMBER

Possible Cause: The script language was expecting a decimal number, but characters other than 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 that .dll. The name of the function cannot be found in the .dll.

Action: Using the *Novell SecureLogin Configuration Guide for Terminal Emulators*, check the configuration for the emulator. Make sure that you typed the HLLAPI function correctly.

-222 BROKER_HLLAPI_DLL_LOAD_FAILED

Possible Cause: Terminal Launcher was unable to load the HLLAPI.DLL that you specified.

Action: Make sure that the path and file that you entered for the .dll are correct.

Possible Cause: The HLLAPI.DLL for that emulator is looking for other .dll files that don't exist or haven't been installed for that emulator. You have probably chosen the wrong .dll file or have specified the wrong HLLAPI function (for example, HLLAPI or WinHLLAPI). Find the correct .dll and function.

You can use Microsoft* Spy++ to find Input and Output IDs.

Action: Check the vendor's documentation for information about that emulator.

-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: Using the *Novell SecureLogin Configuration Guide for Terminal Emulators*, check the configuration for the emulator. Specify the correct emulator type.

-226 BROKER_ERROR_DURING_WINHLLAPISTARTUP

Action: Check the vendor's documentation for information about that emulator.

-227 BROKER_CANNOT_FIND_WINHLLAPICLEANUP_FUNCTION_IN_DLL

Possible Cause: In the Terminal Launcher configuration, you incorrectly specified that the emulator is a WinHLLAPI emulator.

Action: Using the *Novell SecureLogin Configuration Guide for Terminal Emulators*, check the configuration for the emulator. Specify the correct emulator type.

-228 BROKER_BUTTON_NOT_FOUND

Possible Cause: For a Windows single sign-on application, no button exists for the control ID that you specified. For example, if you specified Click #3, no button exists for control ID 3.

Action: Specify the correct control ID number for the button.

-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: The software is not working as intended.

Action: Call Novell Technical Services.

-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.

-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 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, making sure that one restriction doesn't contradict another restriction in the policy.

-242 BROKER_EXCEPTION_CHARACTER_FOUND

Possible Cause: You entered a password that contains a character that isn't allowed.

Action: Use allowable characters in your password.

-243 BROKER_PASSWORD_TOO_SHORT

Possible Cause: You entered a password that doesn't 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: Type 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: The software is not working as intended.

Action: Call Novell Technical Services.

-256 BROKER_UNABLE_TO_GET_NT_CACHE_DIR

Possible Cause: You are using NT 4 Domains mode, but you haven't defined or mapped a Homedrive.

Action: Log in as the user to determine whether the Homedrive and Homepath variables are set. If the variables are not set, use the NT domain administrative tools to set them.

-257 BROKER_UNABLE_TO_CREATE_NT_CACHE_DIR

Possible Cause: The user didn't have rights to create a directory on the user's Homedrive.

Action: Grant the user rights to the directory.

-259 BROKER_MUST_BEGIN_WITH_UPPERCASE

Possible Cause: You entered a password that didn't begin with an uppercase character.

Action: Type an uppercase character at the beginning of the password.

-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.

Action: Install the latest version of SecureLogin.

-264 BROKER_DDE_CONNECT_FAILED

Possible Cause: Terminal Launcher couldn't 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

Action: Refer to the vendor's documentation.

-266 BROKER_NT_FILE_STORAGE_SAVE_FAILED

Possible Cause: Using NT 4 Domains mode, the user was unable to save to the equivalent of a cache file in the Home directory.

Action: Grant the user rights so that the user can write files to the Home directory.

-269 BROKER_NOT_A_PASSWORD_POLICY_COMMAND

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 didn't meet requirements as specified in password policies.

Action: Enter the password correctly.

-273 BROKER_MSTELNET_OPERATION_NOT_SUPPORTED

Possible Cause: The generic emulator can't support a particular operation (for example, SetCursor).

Action: For generic emulators, don't use the command.

-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 that 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 doesn't separate decimal numbers for input and output control IDs.

Action: For generic emulators, you must specify a set of input and output control IDs. Use a comma to separate decimal numbers.

-282 BROKER_INVALID_CHARACTER_FOUND_IN_COPY_ID_LIST

Possible Cause: A comma doesn't 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_TLAUNCH_INI

Possible Cause: SecureLogin is unable to read the TLAUNCH.INI file because the file has been deleted.

Action: Create a blank TLAUNCH.INI file.

Action: Create a 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 has not been defined.

Action: Using Terminal Launcher, specify a terminal type for the emulator.

-290 BROKER_FILE_LOAD_FAILED

Possible Cause: You don't have enough rights to convert an earlier TLAUNCH.INI file to a later format, read an earlier TLAUNCH.INI file, or create a new TLAUNCH.INI file.

Action: The network administrator must assign necessary rights.

-293 BROKER_NO_EXTENDED_TREES_FOUND

Possible Cause: The SecureLogin schema extensions to NDS or eDirectory haven't been installed on any of the NDS trees that you are connected to.

Action: Extend the schema by running SCHEMA.EXE.

Action: Connect to an NDS or eDirectory tree that has the SecureLogin schema extensions.

-294 BROKER_SETPLAT_VARIABLE_MUST_BE_RUN_TIME

Possible Cause: The first argument to a SetPlat argument can be a variable. If it is a variable, it must be a runtime 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 doesn't handle text in the second part of an If command.

Action: Make sure that the command is one listed and documented in the *Novell SecureLogin Administration Guide*.

-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_EXIST

Possible Cause: You have tried to use repeated characters in a password, but a password policy doesn't 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 doesn't 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 doesn't 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_WRITEABLE_VARIABLE

Possible Cause: You tried to save a new value to type of variable that can't be written to.

Action: Use a runtime or normal variable.

-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.

Action: Close the process and try again.

Possible Cause: You don't 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 hasn't been defined.

Action: Define the label.

-317 BROKER_INCORRECT_DATABASE_VERSION

Possible Cause: The version of SecureLogin that you are using doesn't handle the version of SecureLogin that is stored in NDS or eDirectory.

Action: Upgrade to the latest version of SecureLogin.

-318 BROKER_DIRECTORY_CRC_DOES_NOT_MATCH

Possible Cause: Whenever SecureLogin stores an entry in NDS or eDirectory, SecureLogin employs a redundancy check. If the redundancy check doesn't match when SecureLogin reloads the entry, the data in NDS or eDirectory is unusable.

Action: Troubleshoot NDS or eDirectory.

-319 BROKER_DISALLOWED_DUPLICATES_EXIST

Possible Cause: You entered a password that has unacceptable duplicate characters.

Action: Enter the password correctly.

-320 BROKER_GOTO_LIST_ASSERTION

Possible Cause: The software is not working as intended.

Action: Call Novell Technical Services.

-321 BROKER_SUBROUTINE_NOT_DEFINED

Possible Cause: A Call command is calling a subroutine that hasn't yet been defined.

Action: Define the subroutine.

-325 BROKER_ENTRY_MUST_HAVE_NON_NULL_KEY

Possible Cause: You tried to add a script or variable that is a blank string.

Action: Provide a name for the script 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 but isn't.

Action: Change the argument to a variable.

-327 BROKER_OBJECT_NOT_FOUND

Possible Cause: LDAP or Active Directory* can't locate the User object in NDS or eDirectory.

Action: Troubleshoot NDS or eDirectory.

-328 BROKER_ADS_MEMORY_FAILURE

Possible Cause: The Active Directory 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 Active Directory, SecureLogin is unable to read the data.

Action: Troubleshoot Active Directory.

-330 BROKER_ADS_INSUFFICIENT_RIGHTS_TO_DELETE

Possible Cause: When you removed a script, SecureLogin tried to delete part of an attribute from Active Directory. However, you are unable to delete the attribute because you don't have sufficient rights to Active Directory.

Action: The administrator must assign sufficient Active Directory rights for each user so that the user can modify SecureLogin attributes.

-331 BROKER_ADS_ERROR_DELETING_VALUE

Possible Cause: Active Directory was unable to delete a value.

Action: Troubleshoot Active Directory.

-332 BROKER_NO_PASSWORD_FIELD_VARIABLE_IN_SCRIPT

Possible Cause: A Web script must have at least one Type command that has "password" as the second argument. The following lines illustrate a typical script:

```
Type $Username  
Type $Password Password.
```

However, the script 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 didn't 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: There is a problem with the Prot:SSOAuth data attribute.

Action: Call Novell Technical Services.

-340 BROKER_UNKNOWN_DATABASE_VERSION

Possible Cause: You are using an earlier version of SecureLogin.

Action: Upgrade to the latest version of SecureLogin.

-349 BROKER_UNABLE_TO_FIND_SESSION_FILE

Possible Cause: Terminal Launcher couldn't find a session file for an emulator.

Action: Configure Terminal Launcher to have the correct path to the file for the emulator session.

-350 BROKER_ERROR_NO_INDEX

Possible Cause: You started using indexes in a Web script. For example, on the first line you typed the following:

```
Type $Username #1
```

Later in the script, you used a Type command without #2 (or another number). You have mixed index mode and non-index mode.

Action: Either use index mode or non-index mode. Don't mix the two modes.

-353 BROKER_RECURSIVE_SCRIPT_INCLUDE_DETECTED

Possible Cause: While using the Include command, you included a script twice.

Action: Only include a script once.

-354 BROKER_NETWORK_PASSWORD_INCORRECT

Possible Cause: You have turned on the option to prompt the user of the network password before the user can access options on the system tray. The user entered an incorrect password.

Action: Enter the correct password.

-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.

-361 BROKER_NMAS_DLL_NOT_AVAILABLE

Possible Cause: SecureLogin can't load 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 couldn't find the NMAS relogin function in the DLL for NMAS.

Action: Install the latest version of NMAS.

-363 BROKER_STANDARD_VARIABLE_REQUIRED

Possible Cause: The command requires a \$ variable. However, you provided a ? variable.

Action: Provide a \$ variable.

-365 BROKER_LDAP_INIT_FAILED

Possible Cause: The initialization of the LDAP SSL layer failed.

-372 BROKER_ACCESS_IS_DENIED

Possible Cause: For LDAP, you don't have rights to the part of the Directory that you are trying to access.

Action: Grant users 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.

-375 BROKER_NOT_RUNNING_NT

Possible Cause: Although you aren't running NT, you tried to use a feature that is only available through NT.

Action: Don't use that feature unless you are running NT.

-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 Novell Technical Services.

-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 script.

-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 other HLLAPI applications.

-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 Terminal Launcher is set up correctly with the emulator.

-387 BROKER_LAST_NDS_USER_NOT_FOUND

Possible Cause: The last NDS or eDirectory user, as stored in the registry, could not be read for use in an NMAS login.

Action: Make sure that the last NDS or eDirectory user is stored correctly in the registry.

-388 BROKER_LAST_NDS_USER_UNWORTHY

Possible Cause: The last NDS or eDirectory user, as stored in the registry, was not in the correct format. An NMAS login was unable to use the format.

Action: Make sure that the last NDS or eDirectory user 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.

-391 BROKER_LDAP_SSL_ADD_CERT_FAILED

Possible Cause: SecureLogin could not open the certificate you supplied for LDAP over SSL. Either the file doesn't exist or it is in the incorrect format.

If the certificate file specified ends in .DER, SecureLogin uses Distinguished Encoding Rules (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: Make sure that the variable name is correct.

-393 BROKER_SCRIPT_NOT_PURELY_INDEXED

Possible Cause: While working with Web modules, you mix indexed and non-indexed commands. For example, you typed the following:

```
type $username #1
type $password
```

Action: Make sure that all commands use indexes, or remove all indexes. See [“-350 BROKER_ERROR_NO_INDEX” on page 186](#).

-394 BROKER_LDAP_PASSWORD_INCORRECT

Possible Cause: The password supplied to login to LDAP was incorrect.

Action: Check the password.

-395 BROKER_LDAP_USER_NON_EXISTENT

Possible Cause: The username that you used to log in to LDAP does not exist.

Action: Make sure that the username 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.

Action: Make sure that the LDAP server you are connected to is running.

-399 BROKER_DIVIDE_BY_ZERO_IS_BAD

Possible Cause: Using the Divide command, you attempted division by zero.

-400 BROKER_WRONG_SECTION_NAME

Possible Cause: You manually edited a wizard-generated script.

Action: When editing a script, don't 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 script.

-401 BROKER_INVALID_GLOBAL_WIZARD_CONFIG

Possible Cause: You manually edited a wizard-generated script.

Action: When editing a script, don't 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 script.

-402 BROKER_LDAP_ATTRIBUTE_DOES_NOT_EXIST_IN_SCHEMA

Possible Cause: You are running LDAP on eDirectory, but you have not correctly mapped the LDAP attributes.

Action: Check your LDAP attribute mappings. See [“Before Installing SecureLogin with LDAP” on page 21](#).

Possible Cause: You are running LDAP on a platform other than eDirectory. However, the schema is not extended for that platform.

Action: 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.

-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 a 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 username was invalid.

Action: Check the format of the username that you entered.

-412 BROKER_INVALID_OPTION_COMBINATION

Possible Cause: An invalid combination of options was passed to a script command. For example, you passed -right and -raw to the Click command.

Action: Review the documentation for the script 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: For this application, don't use the wizard.

-417 BROKER_ADS_PUT_OCTET_INSUFFICIENT_RIGHTS

Possible Cause: You do not have sufficient rights to ADS to perform the current operation.

Action: Ask the ADS administrator to assign you additional ADS rights.

-418 BROKER_ADS_CLR_OCTET_INSUFFICIENT_RIGHTS

Possible Cause: You do not have sufficient rights to ADS to perform the current operation.

Action: Ask the ADS administrator to assign you additional ADS rights.

-423 BROKER_ERROR_INITIALIZING_DATA_STORES

Possible Cause: SecureLogin was unable to initialize either the primary or secondary datastore.

-424 BROKER_UNABLE_TO_LOAD_SLOTP_DLL

Possible Cause: SLOTP.DLL could not be loaded. This DLL is required for synchronizing one-time passwords 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 or SLNMAS.DLL must be correctly installed for these variables to function.

Action: Make sure that either SLINA.DLL or SLNMAS.DLL is installed.

-430 BROKER_MUST_BE_CALL_OR_GOTO

Possible Cause: When using the OnException command, the second parameter must be either Call or GoTo.

Action: Check the script documentation for OnException. See [OnException/ClearException](#) in *SecureLogin Script Commands*.

Getting Help from Novell Technical Services

Most likely, you won't see the following error codes. If you do encounter one, call Novell Technical Services.

- ◆ -101 BROKER_NO_MORE_COMMANDS
- ◆ -105 BROKER_REMOVE_ENTRY_FAILED
- ◆ -106 BROKER_UPDATE_GET_ENTRY_FAILED
- ◆ -110 BROKER_NO_MORE_PLATFORMS
- ◆ -111 BROKER_NO_MORE_VARIABLES
- ◆ -120 BROKER_INVALID_PREF_DATA_TYPE
- ◆ -121 BROKER_PREFERENCE_DATA_CORRUPT

- ◆ -122 BROKER_TARGET_ENTRY_LIST_NOT_LOADED
- ◆ -129 BROKER_ENTRY_LIST_NOT_NULL
- ◆ -130 BROKER_ENTRY_LIST_NULL
- ◆ -132 BROKER_SYM_LIST_NULL
- ◆ -141 BROKER_PREF_INVALID
- ◆ -142 BROKER_SET_PREF_INVALID
- ◆ -173 BROKER_NO_MORE_CACHE_FILE_DATA
- ◆ -177 BROKER_PUBLIC_KEY_HAS_CHANGED
- ◆ -195 BROKER_FILE_TRAITS_OP_NOT_IMPLEMENTED
- ◆ -196 BROKER_DUMMY_OP_NOT_IMPLEMENTED
- ◆ -200 BROKER_END_OF_SCRIPT
- ◆ -206 BROKER_BREAK_BLOCK
- ◆ -207 BROKER_END_SCRIPT_NOW
- ◆ -210 BROKER_CORPORATE_MOD_ABORTED
- ◆ -213 BROKER_NDS_OP_NOT_IMPLEMENTED
- ◆ -214 BROKER_UNABLE_TO_GET_CURRENT_OU
- ◆ -221 BROKER_HLLAPI_OBJECT_UNINITIALIZED
- ◆ -223 BROKER_HLLAPI_OBJECT_ALREADY_INITIALIZED
- ◆ -231 BROKER_AUTH_CANCEL
- ◆ -234 BROKER_FREE_PLATFORM_SCRIPT_NULL_PTR
- ◆ -235 BROKER_VBA_LOGIN_INTERFACE_NOT_IMPLEMENTED
- ◆ -253 BROKER_NT_FILE_TRAITS_OP_NOT_IMPLEMENTED
- ◆ -260 BROKER_NO_DATA_STORES_LOADED
- ◆ -263 BROKER_DDE_LOGIN_INTERFACE_NOT_IMPLEMENTED
- ◆ -285 BROKER_EMULATOR_INFO_NOT_FOUND
- ◆ -286 BROKER_RELOAD_NOT_ENABLED
- ◆ -287 BROKER_TERMINAL_CONNECT_TRY_AGAIN
- ◆ -289 BROKER_WRONG_OBJECT_TYPE
- ◆ -292 BROKER_DLL_NOT_INITIALIZED
- ◆ -297 BROKER_PARSER_ELSE_STATEMENT_FOUND
- ◆ -311 BROKER_RUN_SCRIPT_AGAIN
- ◆ -312 BROKER_NO_OU_PERIOD_FOUND
- ◆ -320 BROKER_GOTO_LIST_ASSERTION
- ◆ -322 BROKER_UNABLE_TO_FIND_PASSWORD_FIELD
- ◆ -323 BROKER_PASSWORD_FIELD_STYLE_NOT_SET
- ◆ -324 BROKER_WEB_ACTION_NOT_SUPPORTED

- ◆ -337 BROKER_DES_KEY_NOT_SET
- ◆ -338 BROKER_DES_INVALID_BLOCK_LEN
- ◆ -339 BROKER_INVALID_ENCRYPTION_TYPE
- ◆ -341 BROKER_USER_KEY_NOT_SET
- ◆ -343 BROKER_PRIMARY_KEY_DECRYPT_FAILED
- ◆ -344 BROKER_SECONDARY_KEY_DECRYPT_FAILED
- ◆ -345 BROKER_MERGE_WRONG_ENTRY_TYPE
- ◆ -348 BROKER_PASSWORD_RESET_DETECTED
- ◆ -352 BROKER_AUTH_DATA_INCORRECT
- ◆ -355 BROKER_USER_ABORTED_LOAD_PROCESS
- ◆ -357 BROKER_ERROR_REG_CACHE_NO_DETAILS
- ◆ -358 BROKER_ERROR_REG_CACHE_SAVE_FAILED
- ◆ -359 BROKER_ERROR_REG_CACHE_SPLIT
- ◆ -360 BROKER_PASSWORD_VARIABLE_NOT_ALLOWED
- ◆ -364 BROKER_LDAP_LOGIN_CANCELLED
- ◆ -367 BROKER_REG_AUTH_CACHE_MISMATCH
- ◆ -368 BROKER_LDAP_TOKEN_DELETED
- ◆ -369 BROKER_CRED_LIST_NOT_NULL
- ◆ -370 BROKER_CRED_LIST_NULL
- ◆ -371 BROKER_NO_MORE_CRED_SETS
- ◆ -374 BROKER_DUPLICATE_ENTRIES_EXIST
- ◆ -376 BROKER_WINNT_CACHE_AUTH_REG_FAILED
- ◆ -377 BROKER_WINNT_CACHE_AUTH_REG_WRONG_USER.
- ◆ -378 BROKER_INVALID_PIPE_STRING_FOUND
- ◆ -379 BROKER_HEX_LENGTH_INCORRECT
- ◆ -398 BROKER_WIZ_CP_WRONG_SCRIPT_TYPE
- ◆ -408 BROKER_DES_KEY_DATA_CORRUPT
- ◆ -409 BROKER_OPERATION_ABORTED_BY_USER
- ◆ -420 BROKER_SLAASSO_ERR_CRYPTO_KEY_NOT_SET
- ◆ -421 BROKER_SLAASSO_ERR_UNKNOWN_DATA.
- ◆ -422 BROKER_SLAASSO_OUT_OF_MEMORY
- ◆ -427 BROKER_USERNAME_UNSUITABLE_FOR_READING_SLINA_CREDS
- ◆ -428 BROKER_NO_EXCEPTION_HANDLER_DEFINED
- ◆ -429 BROKER_EXCEPTION_RAISED

D

Documentation Updates

This section contains new or updated information on installing and managing Novell® SecureLogin. The information is new since SecureLogin 3.0.3.

This documentation is also provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the documentation changes listed in this section. See [SecureLogin 3.0 \(http://www.novell.com/documentation/lg/securelogin30/index.html\)](http://www.novell.com/documentation/lg/securelogin30/index.html).

If you need to know whether a copy of the PDF documentation you are using is the most recent, check the date that the PDF file was published. The date is in the Legal Notices section, which immediately follows the title page.

New or updated documentation was published on the following dates:

- ◆ “November 21, 2002” on page 196
- ◆ “January 6, 2003” on page 196
- ◆ “January 28, 2003” on page 197
- ◆ “March 21, 2003” on page 197
- ◆ “May 1, 2003” on page 197
- ◆ “October 31, 2003” on page 197

November 21, 2002

Location	Change
“Providing Preset Passphrase Questions” on page 65	Added these topics.
“Allowing Users to Enter Passphrase Questions” on page 66	
“Using a Customized Prompt to Change a Passphrase Question” on page 66	
“Using a Passphrase Policy” on page 67	
“Copying, Exporting, and Importing SecureLogin Settings” on page 68	
“Setting Up Multiple Logins for an Application” on page 85	
“Changing the Startup Order of Applications” on page 87	
“Using Terminal Launcher With Non-HLLAPI-Compliant Emulators” on page 139	Updated the Using SecureLogin Terminal Launcher section to include information on Reflection 9.
.	
Configuration Guide for Terminal Emulators	Added information on 15 emulators.
Script Commands	Updated the Commands chapter and moved it to a separate document (<i>Script Commands</i>).

January 6, 2003

The following updates were made in this section:

Location	Change
“Using Login Watcher” on page 89	Added this topic.
“Lotus Notes” on page 16	Updated the information
“Including Applications for Detection” on page 98.	

January 28, 2003

The Support for Mainframes topic, which mentioned OS/390, was removed from the Overview section. Novell SecureLogin 3.0 doesn't have an OS/390 component.

March 21, 2003

The following updates were made in this section:

Location	Change
Chapter 4, "Adding Applications for Single Sign-On," on page 55	Added this chapter, to provide details about adding Windows applications and Web pages for single sign-on.
Appendix A	This appendix was deleted. The information on terminal emulation is in Configuration Guide for Terminal Emulators .

May 1, 2003

The following updates were made:

Location	Change
"Before Installing SecureLogin with LDAP" on page 21	Replaced an incorrect graphic that referred to the Prot:SSO Entry attribute. Also corrected information on copying a Trusted Root certificate.

October 31, 2003

Location	Change
"Setting Up Multiple Logins for an Application" on page 85	Revised this topic to correct a documentation defect.
Chapter 8, "Setting Up Terminal Emulation," on page 117	Revised the chapter to include information on how to set up emulators by type (HLLAPI, WinHLLAPI, generic, advanced generic).
Appendix A, "Finding Control IDs and Offsets," on page 157	Added this appendix.
Appendix B, "Finding HLLAPI Types," on page 169	Added this appendix.

