SecureLogin Test Application Definitions

About This Document

This document provides information on how to use SecureLogin Test Application. It allows Administrators and other application definition writers to practice their application definition creation skills.

Using Test Application

IMPORTANT: Test Application is intended for administrators and application definition writers for demonstration and test purposes only.

Test Application is available to download at SecureLogin Documentation (https://wwwtest.netiq.com/documentation/securelogin-86/) under **Additional Resources**. This application is designed to replicate an application logon panel and supports the following processes:

- Initial log in
- Wrong password
- Password change

The following example, application definition for the Password Test application, further explains the SecureLogin application definition principles.

Example Application Definition for the Test Application

The application definition for the PSL Password Test Application (`PasswordTest.exe`) provides an example of a typical Windows application definition, including error handling and changing the password. Remember, the password for this application is hard-coded to single when the application is closed and restarted. This can cause confusion when setting strong password policies and changing passwords. You must also create a password policy called PwdTestPolicy, according to the password policy defined in this application definition. The password policy must require a minimum of 6 characters, but no complex rules, in order to use single as a password.

Here is the sample application definition in its entirety. Following this application definition is the explanation of what each section does.

```
# Set Password Policy
RestrictVariable $Password PwdTestPolicy
# ==== BeginSection: Log on ====
Dialog

   Class "#32770"

   Ctrl #1001

   Title "Log on"

EndDialog

SetPrompt "Username =====>"
Type $Username #1001
SetPrompt "Password =====>"
Type $Password #1002
```

```
SetPrompt "Domain =====>"
Type $Domain #1003
Click #1
SetPrompt "Please enter your user name and password to access Password Test.
SecureLogin will remember and automatically log you on in future.  IT Help Desk
x4532"
# ==== EndSection: Log on ====

# ==== BeginSection: Log on failure ====
Dialog

   Class "#32770"

   Title "Log on failure"

EndDialog

# Read the error message and set it as a temporary variable, then clear it
ReadText #65535 ?ErrorMessage
Click #2

# If log on failed, display the current stored Username and Password and prompt the
user to verify them, then retry log on
If "You have failed to log on." -In ?ErrorMessage
   DisplayVariables "Log on to Password Test failed. The password for this
application must be single when it first starts. IT Help Desk x4532"
# Press Alt>F and L to invoke the logon box so the user doesn't have to.

   Type -Raw "\Alt+F"

   Type -Raw "L"

   Type $Username

   Type $Password

   Type $Domain

EndIf

# ==== EndSection: Log on ====

# ==== Begin Section: Change Password ====
# Change Password Dialog Box
Dialog
Class "#32770"
Title "Change Password"
EndDialog

# Backup password, fill in the old user name and password, then start the change
password routine
Set ?PwdBackup $Password
Type $Username #1015
Type $Password #1004
ChangePassword ?NewPwd "Please enter a new password for this application."
Type ?NewPwd #1005
Type ?NewPwd #1006
Click #1
```

```
# Change password successful message
Dialog

   Class "#32770"

   Ctrl #65535 "You have changed your password successfully."

   Title "Change successful"

EndDialog

# Clear application owned message and accept new password
Click #2
Set $Password ?NewPwd
# ==== End Section: Change Password ====
```

Application Definition Explained

You can use the same application definition to show what function each section performs. `Dialog/EndDialog` blocks define a Windows dialog box. When the dialog box appears, SecureLogin detects that this dialog box is based on the information found within the dialog block. The `Dialog/EndDialog` block must contain enough information for the block to be unique, or the application definition runs when other dialog boxes owned by the same executable with the same information appear.

When SecureLogin detects that all the information between `Dialog` and `EndDialog` is contained in the dialog box on the screen (for example, the application login box, the change password box, or the failed logon box), it runs the application definition commands until it sees the next dialog statement or the end of the application definition, whichever is applicable. The order does not matter in Windows application definitions, because SecureLogin watches for all dialog boxes while the executable is running. Use a logical order for troubleshooting purposes.

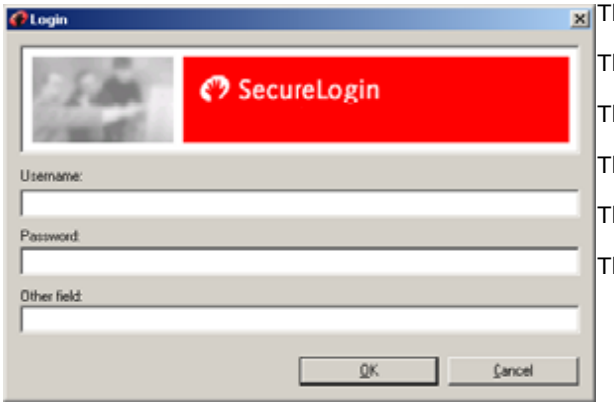
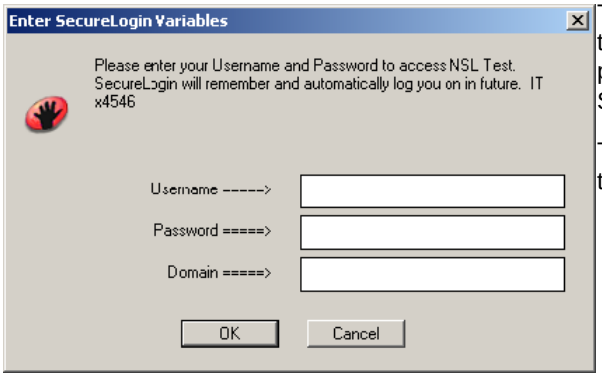
Dialog boxes

The following application definition example shows screen captures of the relevant dialog boxes. You can use the Window Finder tool to gather information about the title of the window, class names, dialog IDs, and so on. Use the wizard to automate the application definition creation.

Application definition section	Comments
`# Set Password PolicyRestrictVariable $Password PwdTestPolicy`	This restricts the $Password variable to comply with the Password Policy "PwdTestPolicy".
`# ==== BeginSection: Log on ====Dialog Class "#32770" Ctrl #1001 Title "Log on"EndDialog`	When `PasswordTest.exe` runs, SecureLogin watches for dialog boxes that appear and match the information defined between the `Dialog/EndDialog` commands. You can specify all values, or a few, as long as the information specified is unique to that dialog box.

Application definition section	Comments

`SetPrompt "Username =====>"`
`"Type $Username #1001`
`SetPrompt "Password =====>"`
`"Type $Password #1002`
`SetPrompt "Domain =====>"`
`Type $Domain #1003`
`Click #1`
`SetPrompt "Please enter your Username and Password to access NSL Test. SecureLogin will remember and automatically log you on in future. IT Helpdesk x4546"`
`# ==== EndSection: Log on ====`
	Type the stored ($) Username variable into #1001, and so on. `SetPrompt` is used to customize the window the user sees when there are no credentials stored.
	
	When the user first runs an application that is newly enabled for single sign-on, SecureLogin prompts for their login credentials, and stores and remembers them for future login attempts.

	The title is Log In. The Class is #32770. The **Username** field is Control ID #1001. The **Password** field is Control ID #1002. The **Other field** is Control ID #1003. The **OK** button is Control ID #1.
	This dialog box is only displayed the first time the application definition is run by a user. It prompts the user to enter credentials for SecureLogin to store. The `SetPrompt` command is used throughout the example application.
	This is the login failure dialog box. The title is Login Failure. The class is #32770. The **OK** button is Control ID #2. The error message is Control ID #65535

Application definition section	Comments
	This is the Change Password dialog box. The **Username** field is Control ID #1015. The **Old Password** field is Control ID #1004. The **New Password** field is Control ID #1005. The **Confirm New Password** field is Control ID #1006. The **OK** button is Control ID #1.
	The `ChangePassword` command is used in the example application definition to display a dialog box for the user to enter a new password. The dialog box is customized to provide more information for the user.