

BNX Authenticated Sign-On™

Version 5.5

Adapter Builder Guide

Copyright Notice

BNX Systems Corporation

1953 Gallows Road, Suite 500

Vienna, VA 22182

Phone: 703-734-9200 Fax: 703-734-6553

Web Site: www.bnx.com

Sales email: sales@bnx.com

© 2005 BNX Systems Corporation. All rights reserved.
No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without written permission from the publisher.

BNX, BNX Systems Corporation, BNX Authenticated Sign-On, BAS Administration Manager, and BAS Database Manager are pending trademarks, registered trademarks, or service marks of BNX.

All other product names mentioned herein are the trademarks of the product owner.

BNX Adapter Builder™ Guide

Table of Contents

Chapter 1 How to Use This Guide	1
Who Should Use This Guide	1
How to Use This Guide.....	1
How to Find Related Documentation	2
How to Get More Help	3
Chapter 2 Overview of the Adapter Builder	4
Installation and Distribution of the Adapter Builder.....	4
The Adapter Builder Wizard.....	4
Comparing the Adapter Builder Express Mode to Advanced Mode	5
Selecting the Application Type	6
Advanced Mode Concepts.....	6
Task Types	6
Tasks, Subtasks, Activities, and Actions	7
Application States	8
Chapter 3 Building Adapters	9
Building Adapters Using Express Mode	9
Building Adapters Using Advanced Mode	9
Before Building Your Adapter	9
Steps in the Advanced Mode Process.....	11
Configuring Application States and Match Criteria	12
Scripting Actions	13
Chapter 4 Modifying Existing Adapters	14
Modifying an Adapter in Express Mode	14
Modifying an Adapter in Advanced Mode.....	14
Using the Tree View.....	14
Editing Startup Information	15
Important Configuration Options.....	15
Modifying Substate Elements	15
Other Configuration Options	16
Chapter 5 Troubleshooting	18
Chapter 6 Glossary	20

Chapter 1 How to Use This Guide

Who Should Use This Guide

This guide is intended for those responsible for creating Single Sign-On (SSO) adapters that interface between BNX software and application login dialog boxes and other graphic elements. This guide requires working knowledge of basic Windows functionality and of BNX terminology and software.

How to Use This Guide

This guide focuses primarily on conceptual information about the Adapter Builder. Specific tasks, procedures, and GUI reference information are documented primarily in the Adapter Builder help system.

The following table contains quick links to useful sections of this document:

To . . .	See . . .
Understand and launch the Adapter Builder wizard	The Adapter Builder Wizard (page 4)
Decide whether to use the Adapter Builder Express Mode or Advanced Mode	Comparing the Adapter Builder Express Mode to Advanced Mode (page 5)
Learn Express Mode concepts and operations	Selecting the Application Type (page 6) Building Adapters Using Express Mode (page 9) Modifying an Adapter in Express Mode (page 14)
Learn Advanced Mode concepts and operations	Advanced Mode Concepts (page 6) Building Adapters Using Advanced Mode (page 9) Modifying an Adapter in Advanced Mode (page 14)
How to deploy adapters for use in your BNX Authenticated Sign-On™ environment	Deploying Adapters (page 16)
Troubleshoot adapters	Troubleshooting (page 18)
Learn key terms	Glossary (page 20)

How to Find Related Documentation

This is a list of product documentation specifically and generally relevant to the Adapter Builder.

All BNX documentation is located on the product installation CD and can be manually copied to any location you wish. During product installation, you have the option to install the documentation on your local hard drive. If you choose this option, the documentation can be accessed by clicking **Start > Programs > BNX > BNX Documentation**. Help systems are installed with the appropriate product and can be accessed from within the software's runtime environment, both from the Help menu and from the context-sensitive help buttons (or the F1 key) in the wizard.

To learn . . .	See . . .	File name
Concepts, capabilities, best practices, and general information about the Adapter Builder	This document	adapterbuilderguide.pdf
How to perform specific runtime procedures and understand the graphic user interface of the Adapter Builder	Adapter Builder Help	bnxadapbuild.hlp
Only the basic, critical steps necessary to get started using BNX Software	BNX Authenticated Sign-On™ with SQL Server QUICK START GUIDELINES	sqlquickstartguidelines.pdf
	BNX Authenticated Sign-On™ with Active Directory QUICK START GUIDELINES	adsquickstartguidelines.pdf
How to install BNX Software	BNX Software Installation Guide	bnxinstallguide.pdf
Concepts, capabilities, best practices, and configuration information for administrators of BNX Software.	BNX Authenticated Sign-On™ with Active Directory Administration Guide	bnxadsadminguide.pdf
	BNX Authenticated Sign-On™ with SQL Server Administration Guide	bnxsqladminguide.pdf
Concepts, capabilities, best practices, and general information for end users of BNX Software.	BNX Authenticated Sign-On™ End User Guide	bnxenduserguide.pdf

How to Get More Help

For additional help, please contact BNX Customer Care using one of the following methods:

E-mail: support@bnx.com

Website: http://www.bnx.com/web/corporate/cust_care.htm

Telephone support:

Business Hours

Monday-Friday

8:30 a.m. - 5:30 p.m.

800-397-7561

Off-Hours Customer Care

Available 24 hrs./day

800-397-7561 x203

Chapter 2 Overview of the Adapter Builder

Application adapters link BNX Sign-On technology to your applications without any modifications to the applications themselves. They automate actions a user takes to launch an application, log in, navigate, log out, and close the application. When a user starts a BNX Sign-On session and launches an application, the adapter detects specific application states and performs scripted actions, such as entering a user name and password, clicking a button, or selecting a menu item. Information such as user names and passwords are retrieved from the BNX Server (or local cache if enabled.)

The BNX Sign-On Adapter Builder is a tool to create adapters for Windows, Web, and Terminal Emulator (TE) applications. The administrator can tailor an adapter to users' needs. For example:

- A Web adapter navigates to a logon page using Internet Explorer, enters the user name and password, and navigates to an area on the home page where the user typically begins working.
- A Windows adapter logs the user into an application, opens a specific window, and selects a menu item.
- A TE adapter connects to a mainframe via a telnet session, logs in the user, and enters common parameters to navigate to a specific location.

Installation and Distribution of the Adapter Builder

The Adapter Builder is installed automatically with the BNX Administration Manager. It can also be installed separately from other BNX components. Separate installation offers numerous advantages.

First, it allows BNX administrators to minimize their installation footprint. An administrator may need to create adapters on multiple machines, but only use a single machine for administration.

Second, the administrator can delegate adapter creation responsibilities to end users without giving them administrative privileges. Often these users are more familiar with specific applications and can create robust adapters more efficiently and easily. For example, an end user could create basic web adapters for logging into simple intranet pages. Another user could specialize in creating complex adapters for legacy console applications, and so on. However, administrator privileges are required to deploy adapters in the BNX system, so the administrator retains ultimate control of the process.

The Adapter Builder Wizard

The Adapter Builder employs a wizard to guide you through the process of creating adapters. To start the wizard, select **Start → Programs → BNX → Adapter Builder**.

The Adapter Builder wizard has two modes, Express Mode and Advanced Mode. Express Mode is designed to create and edit simple Windows and Web adapters with basic functionality. Advanced Mode is used to design and edit more sophisticated adapters. Consult the following table for a detailed comparison of the two modes.

Comparing the Adapter Builder Express Mode to Advanced Mode

If you need to ...	You can use the following mode(s):	
	Express	Advanced
Create simple login adapters for Windows or Web applications.	✓	✓
Create an adapter that <i>only</i> manages login, failed password, and changed password dialog boxes.	✓	✓
Execute simple input into text fields and single button clicks.	✓	✓
Select application menu options, radio buttons, or check boxes.		✓
Submit control keys (e.g., Enter, Alt-F, F11).		✓
Create adapters for Terminal Emulator applications.		✓
Work with applications that include a combination of Web and Windows controls.		✓
Handle pre-or post-login navigation to additional application screens, windows, or web pages (for example, navigating to a login page after opening an initial web page).		✓
Log out or close the application.		✓
Handle applications that launch a secondary process or executable.		✓
Handle dynamic class IDs or subtle changes in the application state.		✓

Selecting the Application Type

When you launch the Adapter Builder wizard and begin to create a new adapter, you must select the application type. The following table lists aspects that identify each application type:

Application Type	Description
Web	Launched through the Internet Explorer web browser. Note: Other web browsers are not supported. Can include graphic user interface (GUI) elements and controls such as menus, buttons, and list boxes.
Windows	Non-browser-based application. Relies heavily on GUI elements and controls.
Terminal Emulator (TE)	Accesses an application through a console that contains only text, such as a telnet session. Console window employs no secondary windows or GUI elements.

Advanced Mode Concepts

This section discusses the following topics, which are essential for users who need to create or modify advanced adapters:

- Task types
- The hierarchy and definitions of tasks, subtasks, activities, and actions
- Application states and substates

Task Types

Tasks are sequences of user activities that produce a usable result. BNX adapters support the following types:

Prelogin – Actions that occur before the user logs in to the application, such as selecting a menu item to invoke a login dialog.

Login – Actions executed when the user logs in to the application, such as entering a user name and password and clicking a button to submit the information.

Postlogin – Actions executed after the user logs in to the application; for example, selecting a menu item or navigating to another part of the application.

Force Password Change – Actions executed when BNX Password Rules applied to the application force the user to change his or her password.

Logout – Actions executed when the user logs out without shutting down the application. This task is initiated from the Logout Applications window, which is invoked from the BNX system tray menu on an SSO client.

PreTerminate – Actions executed to gracefully close the application. If this task is not configured, the system abruptly terminates the running application process when the user selects “Shut Down BNX Sign-On Session and Close Applications” from the BNX system tray menu.

Tasks, Subtasks, Activities, and Actions

These are the building blocks of BNX adapters. They have the following hierarchy:

Tasks >
 Subtasks >
 Activities >
 Actions

In order from simple to complex:

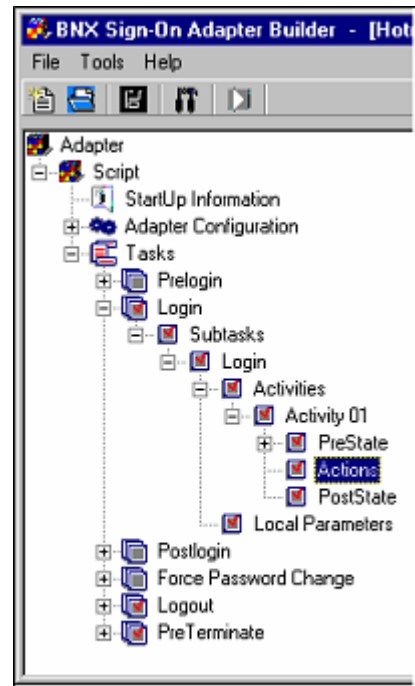
An **action** is a discrete user action against a specific element, such as typing a string in a text field, pressing a control key, clicking a button, or selecting an item from a drop-down list.

An **activity** combines an **application state** and one or more actions. For example, when a confirmation dialog box appears (the application state), the user clicks the **OK** button (the action). An activity can include a single action such as clicking a button, or multiple actions, such as entering a user name and password in text fields and clicking **OK** to submit the data. See below for examples.

See *Application States* on page 8 for more information on application states.

A **subtask** is a collection of one or more activities. It usually contains only one.

A **task** is a logical grouping of one or more subtasks.



Task hierarchy illustrated in advanced editing mode

Note: Limiting a subtask to a single activity usually provides better performance and makes the subtask easier to troubleshoot and maintain. No poststate is required for single-activity subtasks, which reduces the overhead required for matching application states.

However, occasionally two activities must be chained together. For example, in a subtask to handle a password change, the activity of entering the old password must be chained to the activity of entering the new password in a new dialog window. In this case, the password-change subtask contains two activities, and you must record a poststate after the actions of the first activity are executed. The prestate of the second activity must match the poststate of the first activity, linking the activities.

Examples

The following is an example of two single-activity subtasks, each with a single action:

```
PreTerminate task:  
    Subtask 1:  
        Activity 1:  
            Prestate = Main window  
            Action 1 = Select Exit from the File menu  
    Subtask 2:  
        Activity 1:  
            Prestate = Confirmation dialog window  
            Action 1 = Click OK to dismiss the dialog
```

The following Login task contains one single-activity subtask with multiple actions:

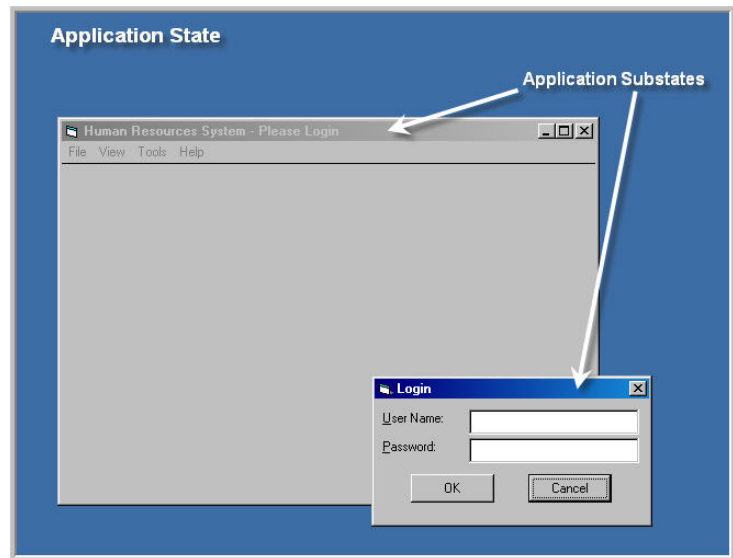
```
Login task:  
  Subtask 1:  
    Activity 1:  
      Prestate = Login dialog  
      Action 1 = Enter user name  
      Action 2 = Enter password  
      Action 3 = Press Enter
```

The following password-change subtask contains two activities, with a poststate after the first:

```
Subtask Password Change:  
  Activity 1: Enter Old Password  
    Prestate = Enter Password dialog  
    Action 1 = Enter password  
    Action 2 = Click OK  
    Poststate = Enter New Password dialog  
  Activity 2: Enter New Password  
    Prestate = Enter New Password dialog  
    Action 1 = Enter password  
    Action 2 = Click OK
```

Application States

An **application state** is the set of window properties (e.g., the window title or other text or controls in the window) that identifies a particular point in the execution of the application. For example, the title of a login dialog and the text prompting the user to enter a password might define a specific state in that application's execution. A BNX adapter records the application state required for each scripted action. When it detects that state, it executes the action.



Each state includes at least one **substate**. If an application is composed of multiple windows (for example, a main window accompanied by a popup login dialog box), each window is in effect a distinct substate. For an application state to be accurately detected at runtime, you must capture each substate and configure its match criteria.

A **prestate** is an application state at the beginning of an activity—the specific application state that must be detected at runtime in order to trigger the scripted actions. You must define a prestate for each activity. In a few cases, a **poststate**—the application state after the scripted actions are executed—is also required.

Chapter 3 Building Adapters

Building Adapters Using Express Mode

The process of building simple adapters using Express Mode includes the following activities:

1. Select the application type. (Express Mode handles only Web or Windows applications.)
2. Name the adapter and describe the path to the adapter file.
3. Define login behavior for the adapter.

Optional steps:

- Define how the adapter handles password failure.
- Define how the adapter handles password change requests.

Building Adapters Using Advanced Mode

Before Building Your Adapter

It is useful, while planning your adapter, to carefully observe the way your application actually functions in light of the concepts detailed in *Advanced Mode Concepts* on page 6. We recommend you take some or all of the following steps before starting to create an adapter.

- Run the application the way it would normally run from the client machine. Note whether this process differs across client machines.
- Use the Windows Task Manager to see whether the application has secondary executables, i.e., whether it spawns other processes when it executes. If so, do they run concurrently with the application, or does the application shut down after they are launched? **Note:** To see the process, right-click on the application in the Applications tab and select Go to Process from the context menu.
- Execute the sequence of actions the user typically takes from the time the application is launched to when the application process is terminated. Note the unique identifiers in each sequence, such as text in the window title, the class names, or the control IDs.
- Go through the atypical sequence of actions the user may encounter outside the normal application runtime behavior—for example, error conditions such as password failure or intermittent occurrences such as password expiration.
- Sort these actions into the task categories and then into subtasks.

Timing Issues

Timing issues can impact the robustness and reliability of an adapter. Such issues can be complicated by variations in individual machines, different user profiles, and other factors impacting application behavior. For example, high network traffic can cause a website login to last an unusually long period of time, which can cause your adapter to fail if it does not adjust for this possibility. In order to ensure that your adapter's actions synchronize with real-time application execution, consider the following questions:

- Does the timing of application execution vary each time the application is run? Test various scenarios.
- Is there a consistent and significant amount of time between an executed action and the resulting change in the application state?
- Is this gap machine-dependent or independent?

Not all timing problems can be resolved through adjusting adapter settings. Sometimes the network or hardware environment must be adjusted before adapters can function reliably. However, in a situation where timing lags are consistent, they can usually be addressed through two methods:

1. The ***State Search Max Wait Time*** value specifies, in seconds, how long the adapter will try to match a subtask prestate at runtime. The default is 5 seconds. On slower machines or applications, you may need to raise this value. If you adjust this value, you must also adjust the ***State Search Retries*** value to maintain a retry interval of 0.5 seconds.
2. ***Insert Pause*** after a specific action in the script. Use this method sparingly, i.e., only when the first method does not suffice.

Steps in the Advanced Mode Process

The following outline lists the basic steps required to build an adapter in Advanced Mode. See the Adapter Builder Help for details about these steps.

1. Define adapter startup information.

- Describe the adapter and where you want to store it on your file system.
- Define the location of the web page or executable file.
- Enter other adapter-specific information as necessary.

2. Select tasks.

Select the adapter task(s) you want to define.

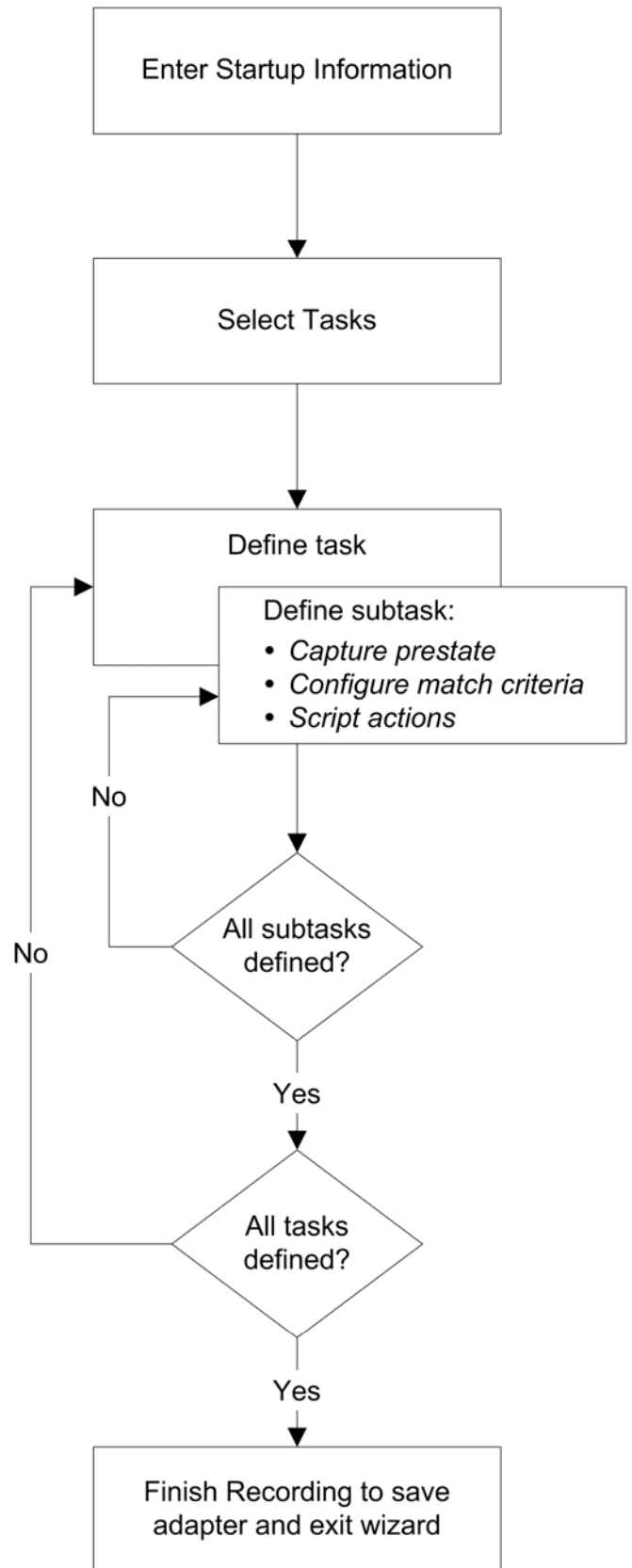
3. Record and configure tasks/subtasks.

For each subtask of each selected task, capture the application state and script the actions the adapter should execute when it detects that application state at runtime.

Note: As a general rule, create a new subtask each time the application state changes, i.e., each time the web page, window, or console screen changes:

- Drop the Adapter Builder Detector icon on the page, window, or screen.
- For each substate, select elements and clarify the match criteria. For example, a web page may have text elements that change dynamically according to the context. You can use match criteria to adjust the adapter's sensitivity to such changes.
- Script the specific actions to be taken, such as entering a user name and password and then clicking a button or link.

- **Warning:** If you make a mistake recording a subtask, you cannot undo the mistake in the wizard interface. Complete the adapter, save it, and then modify or re-record the subtask using the Adapter Builder advanced editing mode. For more information on re-recording subtasks, see *Modifying an Adapter in Advanced Mode* on page 14.



Configuring Application States and Match Criteria

This section describes the method and considerations for configuring application states and criteria. It covers the following topics:

- Common steps for all adapters
- Considerations for Web adapters
- Considerations for Windows adapters
- Considerations for TE adapters

Common Steps for All Adapters

1. Drop the Detector icon on a window or screen and click **Next**. The State Configuration window appears.

Note: Change the **Reason for match** field only in the case of password failure or expiration.

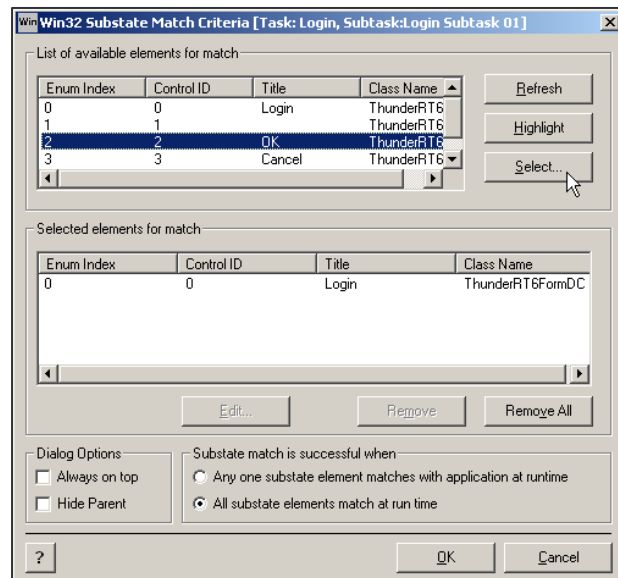
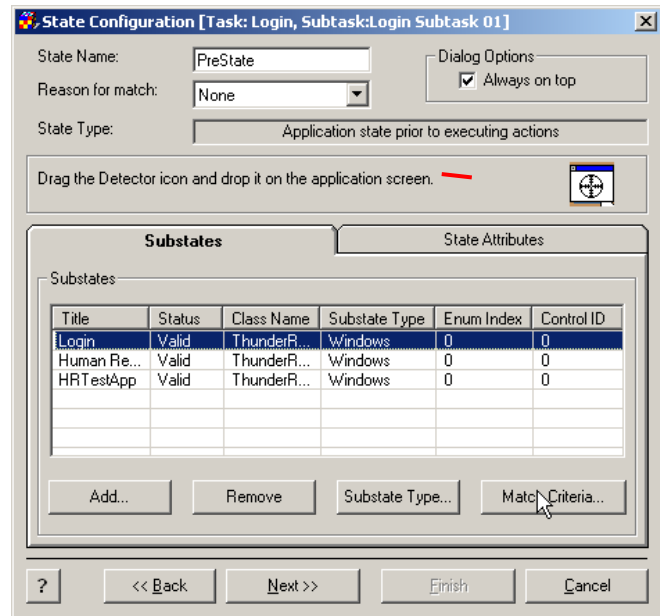
2. Select a substate and click **Match Criteria**. The Substate Match Criteria window appears.

3. Select elements from this window to be used in matching the substate at runtime.

For details about using these windows, see the Adapter Builder Help.

Considerations for Web Adapters

Some applications use Internet Explorer and a URL but are not true Web applications. For example, they may be collections of ActiveX controls rather than HTML. In such cases, select Web as the *adapter* type when creating the adapter. However, select Windows as the *substate* type in the State Configuration window. You may need to experiment with both Windows and Web to see which substate type correctly captures the window elements.



When multiple top-level windows are captured for Web applications, a single substate is added to the Substates list. However, in that scenario, multiple substates should exist. Manually add and configure the additional substates properly, or an error will occur.

Considerations for Windows Adapters

To configure the state-match criteria for Windows applications, generally use the title bar or class name of the window itself (usually the first item in the Element list) or a piece of text in the window that uniquely identifies it.

Note: In addition to allowing a partial match, you will sometimes need to truncate the captured class name or text/title to create a robust adapter.

Considerations for TE Adapters

When multiple top-level windows are captured, all corresponding substates will be Console types. However, in that scenario, only one will truly be a Console substate; the other(s) will be Windows. Manually configure each substate properly, or an error will occur.

Scripting Actions

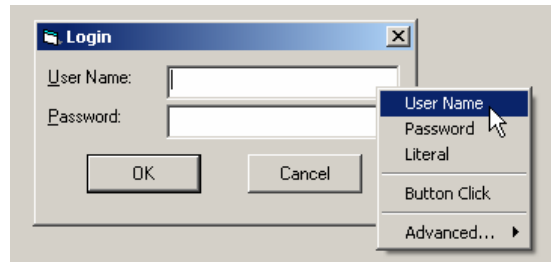
This section details the most common adapter actions. This process is the same for all application types; only the types of actions you script differ.

Scripting an action has two parts:

- Select a window element, such as a text-entry field, menu, button, hyperlink, or TE screen.
- Script one or more actions against that element.

There are two methods for scripting actions:

1. Right-click an element in the application and choose an action from the popup menu, such as typing a user name or clicking a button.
2. Select an element from the Target Element List in the **Configure Script Actions** window (shown in second screenshot at right) and click **Add Action**.



Scripting Actions Using Right-Click

Considerations for Web Applications

You can list the elements as Windows or Web Elements. Enumerate Windows Elements if ActiveX Elements are present on the web page.

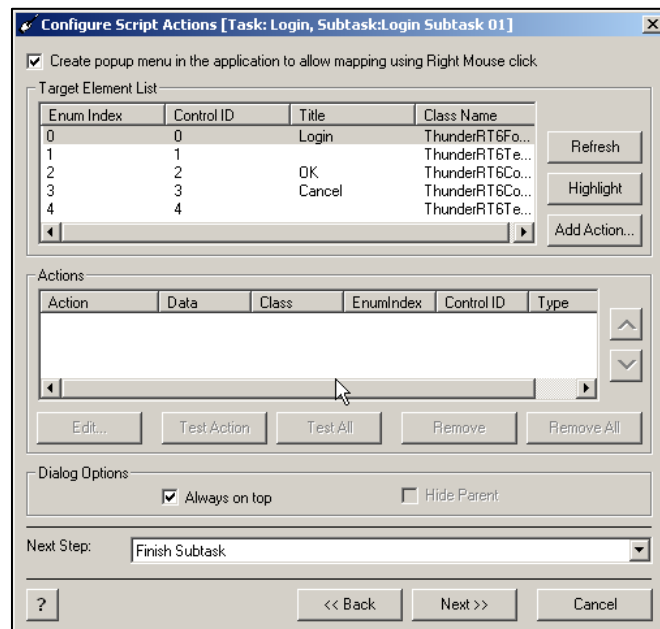
Considerations for Windows Applications

Some nonstandard Windows controls may not be not supported.

Considerations for TE Applications

HLLAPI Mode: Use the **Send Using HLLAPI** method to send text and special commands to the TE application.

Legacy Mode: For some adapters created in the TE Legacy mode, you may need to insert a small pause at the end of a console activity. This gives the screen time to redraw before the adapter attempts the next state match.



Chapter 4 Modifying Existing Adapters

To modify a previously saved adapter, launch the Adapter Builder from **Start > Program Files > BNX > Adapter Builder**. The Adapter Builder wizard opens. Select the **Modify** tab and open the adapter you want to modify from the list.

This chapter details how to configure and troubleshoot existing adapters. It covers the following topics:

- Modifying adapters in Express Mode
- Modifying adapters in Advanced Mode
- Using the tree view
- Editing startup information
- Modifying substate elements
- Controlling other configuration options

Modifying an Adapter in Express Mode

You can modify an adapter created in Express Mode using the same Express Mode wizard.

To modify an adapter using Express Mode, start the Adapter Builder and select the **Modify** tab, then select the **Express** option. The Adapter Builder wizard will launch in Express Mode. Make any desired changes to the adapter using the wizard.

Note: You can use Advanced Mode to modify an adapter created in Express Mode if you wish to add functionality that is not supported in the Express Mode. However, once an Express Mode adapter is saved using Advanced Mode, it can no longer be edited in Express Mode.

Modifying an Adapter in Advanced Mode

If you select **Advanced**, the Adapter Builder opens the adapter in the tree view editing window. You can change advanced adapter components using this interface, including the Startup Information, Configuration Options, and the task(s) you recorded.

Using the Tree View

You can modify, re-record, and delete tasks and their subordinate elements in the Adapter Builder tree editing window. The editing window features two panes, as illustrated in the figure to the right.

Name	Value
StateHandler Info	
Parameter Qualifiers	
Substates	
Name	PreState
Disable State	False
Timeout	20
Reason for Match	None
Number of Retries	3
Possible Destination	
State Match Criteria	All substates match
Node ID	57

Left-clicking on an element in the tree displays its properties grid in the right-hand pane, where editable properties can be modified. Right-clicking on an element in the tree allows you to select other options from a context menu.

Some grid values are frequently useful for modifying or troubleshooting an adapter. Others are useful in special cases, and some can be ignored altogether. We will discuss the most important startup, configuration, and subtask values first, followed by ones used less frequently. If a grid value is not mentioned in BNX documentation, it is because it is no longer useful or should not be changed.

Note: Whenever you modify an adapter, you must reload it into the BNX Sign-On Profile to enable the changes at runtime.

Once you have recorded and saved an adapter, you can do the following:

- Right-click on any task to re-record the entire task.
- Right-click on the Subtasks branch to record a new subtask.
- Right-click on any subtask to re-record or delete the subtask.
- Right-click on any recorded activity to delete it.
- Right-click on Actions to modify or remove recorded actions.
- Left-click on any item to view its fields and modify configurable fields.

Editing Startup Information

When you create an adapter, you define startup information for the adapter and the application. You can edit most of these values in the grid.

Important Configuration Options

The following configuration options are the most important in tweaking and troubleshooting adapters:

- StartType
- State Search Max Wait Time
- State Search Retries

StartType

This value tells the adapter to treat an application as a single instance or as a multiple instance during runtime. The **MultipleInstance** value allows user to close and relaunch an application multiple times within an SSO Session. The **SingleInstance** value allows the user to launch the application only once within an SSO Session.

State Search Maximum Wait Time

This value specifies, in seconds, how long the adapter will try to match a subtask prestate at runtime. The default is 5 seconds. On slower machines or applications, you may need to raise this value.

State Search Retries

This value specifies the number of times the adapter will try to match a subtask prestate at runtime. The default is 10 retries. If you raise the State Search Max Wait Time value, raise the State Search Retries value proportionately to maintain the ratio of 2 retries per second.

Modifying Substate Elements

The most frequently tweaked fields are those for match criteria and substate elements. You may need to alter one of the match-criteria settings. You can also change the substate name.

For example, in certain applications the class name or title bar changes dynamically. It is frequently good practice to truncate these elements and allow a partial match to ensure a successful match. “Allow partial match” is the default setting, but if you forget to truncate an element when configuring the substate, you can select it in the tree and modify it in the corresponding grid.

Other Configuration Options

Secondary Executable

This value specifies the name and the location of a secondary application process that runs concurrently with, or is launched after, the initial executable or batch file starts.

To add a secondary executable to the adapter, right-click on Secondary Executables and select **Add Secondary Executable**. Add the executable information to the name and value fields.

Note: If you add a secondary executable, be sure to change the **Process Match Logic** setting to New Process Match or Multi Process Match.

Process Match Logic

This value sets the process match type of the application. It has the following values:

Same Process Match: the application operates with a single process. This is the default value. Change it only when the application has a secondary executable.

New Process Match: the application executes a new process after shutting down an initial process. You must use this setting if you are using a batch file or application launcher to launch the application

Multi Process Match: the application executes multiple, concurrent processes. Note that Multi Process Match attempts to perform state matching by querying all currently running processes. This type of state matching is slower than the other types. Whenever possible, use Same Process Match or New Process Match for state matching.

WinHLLAPI and WinHLLAPIDLL

These values specify the previously configured WinHLLAPI package and the location of the WinHLLAPI dll for TE adapters. Change them only if the name or path of the WinHLLAPI dll files changes, or to modify an adapter for use with another TE package.

Prestate Top-Level Values

On rare occasions, you may need to edit one of these fields:

Timeout and Number of Retries: You can raise these values if the global State Search Max Wait Time and State Search Retries are insufficient to match a prestate for an activity. These values should rarely be changed.

Reason for Match: If you forget to change the Reason for Match when building a Password Failure or Password Expiration subtask, you can change the value here.

State Match Criteria: You can change this value to “Only Configured Substates Are Present in the Application at Runtime” when you need to limit a match to specified windows, i.e., if any other windows are present, the match will fail. “Any one substate matches with application state at runtime” should not be needed if the substates are configured correctly.

Deploying Adapters

Before an application adapter can be used in a BNX Sign-On session, it must be added to a Sign-On Profile or SSO Adapter Group. Assuming that these elements have already been defined, perform the following steps to deploy a new adapter:

- Logon to the BNX Administration Manager (SQL Server) or Administration Console (Active Directory).
- Define the application to BNX.
- Define user identities for the application.
- Add the application adapter to the Sign-On Profile (SQL Server) or SSO Adapter Group (Active Directory).



For more detailed steps, see the *BNX Authenticated Sign-On with SQL Server Version 5.5 Administration Guide* or the *BNX Authenticated Sign-On with Active Directory Version 5.5 Administration Guide*.

Chapter 5 Troubleshooting

The following table lists common problems encountered when building adapters with the Adapter Builder wizard and running adapters on the client machine. The table also provides detail about the most likely conditions causing these issues and their solutions.

Error	Common Condition	Response
<p>The Adapter Builder wizard fails to detect a window.</p> <p>The Adapter Builder wizard fails to detect a substate.</p>	<p>The Adapter Builder wizard did not identify the application window properly. This error occurs when the state is being captured.</p>	<p>Make sure you drop the icon in the State Configuration Dialog in the title bar of the application window to detect it accurately.</p>
<p>Adapter runtime could not recognize the application state.</p> <p>Application is in a state that has not been recorded in the Adapter.</p>	<p>The launched application reached a state that was not recorded in the adapter. When the adapter was executed, it could not recognize the current screen. These errors occur at client runtime of the adapter.</p>	<p>Launch the Adapter Builder, open the adapter, and review the state match criteria.</p> <p>Does the Windows title need to be truncated?</p> <p>Does the Control ID match the application at runtime?</p> <p>Do the Configuration settings for State Search Max Wait Time and State Search Retries allow enough time for the application to fully launch before the adapter make its last state match attempt?</p>
<p>The system is unable to locate the application to perform the operation.</p> <p>Single Sign-On application launch was unsuccessful because of invalid exe.</p>	<p>The application is absent from the client workstation, or the executable path is incorrect in the adapter. These errors occur at client runtime of the adapter.</p>	<p>Make sure the application exists on the client machine and is located in the path specified in the adapter. Launch the Adapter Builder, open the adapter, and review the Startup Information.</p>
<p>Unable to find requested application-realm in the list of applications.</p>	<p>The SSO application and realm for the application are not defined in the BNX database. This error occurs at client runtime of the adapter.</p>	<p>Make sure you have followed the steps in <i>Deploying Adapters</i> on page 17. In the BNX administration console, you should have an application and realm created for each SSO application.</p>

Error	Common Condition	Response
<p>Specified identity does not exist in the database.</p> <p>Specified identity does not have adapter(s) defined in SSO profile.</p> <p>Specified identity does not have SSO enabled application(s) defined.</p>	<p>The user is incorrectly configured in the database: No corresponding identity exists for the application-realm, no profile is associated with the user, or no SSO adapter is associated with the identity. These errors occur at client runtime of the adapter.</p>	<p>Make sure you have followed the steps in <i>Deploying Adapters</i> on page 17. In the BNX administration console, the user should have an identity for the SSO application and an associated SSO profile.</p>
<p>Cannot launch application; authentication prior to application launch failed.</p>	<p>The user failed secondary authentication required to launch the application. This error occurs at client runtime of the adapter.</p>	<p>Make sure you have followed the steps in <i>Deploying Adapters</i> on page 17. In the BNX administration console, the user should be enrolled with the authentication method defined for the profile.</p>
<p>Adapter file not found.</p>	<p>The sso, xml, or bin file was not found or the metafile has an invalid version number. This error occurs at client runtime of the adapter.</p>	<p>The adapter directory on the client workstation should contain the adapter file and the metafile should contain the correct version number.</p>
<p>Adapter metafile not found.</p>	<p>The metafile that defines the adapter file version was not found on the client workstation. This error occurs at client runtime of the adapter</p>	<p>A metafile by the name "meta<adapter name>.txt" must be created under the cache directory. The metafile should contain the version information of the adapter, in the format "<version>x.x</version>."</p>
<p>Adapter metafile does not contain valid version format.</p>	<p>The metafile that defines the adapter file version does not contain the version number. This error occurs at client runtime of the adapter</p>	<p>The metafile should contain the version information of the adapter, in the format "<version>x.x</version>."</p>

Chapter 6 Glossary

Term/Field	Definition
Action	A combination of an element and a discrete user action against that element, e.g., clicking an OK button.
Activity	A combination of a prestate setting, actions that change the application state, and occasionally a poststate setting.
Adapter	A collection of tasks and adapter configuration settings that define how BNX Single Sign-On software responds to application behavior during runtime. An adapter is created using the Adapter Builder wizard.
Element	A combination of an application window's Enum Index, title, and class name. Elements are listed in the Configure Script Actions window in the Target Element List.
HLLAPI	Windows High Level Language Application Programming Interface; also known as WinHLLAPI.
Insert Pause	A script action that adds an interval of time after actions in the script. Not recommended in most cases.
Literal	A fixed value that the script passes to the application during runtime.
Login task	Task that captures actions that occur when the user logs in to the application. This may include entering a user name (Identity), password (Secret), and domain (Realm).
Logout task	Task that captures actions that occur when the user logs out of the application, without shutting down the application.
Poststate	The application state after the execution of actions. Used infrequently.
Prelogin task	Task that captures actions that occur before the user logs in to the application.
Prestate	The application state prior to the execution of actions.
PreTerminate task	Task that captures actions that occur when the user closes the application.
State/Substate	The current state of an application, defined by elements such as text, window title, parent and child windows, and Control IDs. An application state contains one or more substates. For example, each open window is a substate.
State/Substate Match	Compares the application window elements (title, text, class, etc.) at runtime to recorded elements and criteria to determine whether to execute scripted actions.
Subtask	A collection of one or more activities (usually one).
Task	A collection of one or more subtasks that, combined with adapter configuration settings, compose the script. Tasks include Prelogin, Login, Postlogin, Force Password Change, Logout, and PreTerminate.