

About Using Window and Control Checkpoints

You can create window and control checkpoints in QA Wizard Pro that help verify your tested application is working correctly. Checkpoints verify that attributes of an application exist or are correct when compared to an expected value during playback. This How To walks you through creating window and control checkpoints during recording, or adding them to already existing scripts.

Using Window and Control Checkpoints Overview

A checkpoint fails when a property you are verifying does not exist or match the expected value during playback. You can configure the failure behavior to stop or continue playback and display an error or warning in the Errors pane.

The chart below explains window and control checkpoints in more detail:

Type	Description
Window	Verify the property values of application windows against an expected value. For example, you can verify a window is maximized, has focus, or contains a specific number of objects.
Control	Verify the property values of application controls against an expected value. For example, you can verify a control contains specific text, has focus, or contains a specific number of items.

Checkpoint Examples

The following examples show statements that verify properties of windows and controls. The properties you can use to create checkpoints vary based on the type of window or control you are checking.

Note: The examples show the statements as you would enter them in Text View. You can also use the Add Statement dialog box to add these statements in Grid View.

Verifying window properties

In this example, a checkpoint verifies a window is maximized. The Checkpoint statement verifies the value of the Window Is Maximized property is set to True. If the window is not maximized, the checkpoint fails and an error message is displayed in the Errors pane.

```
Window("WysiCorp CRM").Checkpoint("Window Is Maximized", True, True, "The WysiCorp CRM window is not maximized")
```

Verifying text in a control

In this example, a checkpoint verifies text on a button. The Checkpoint statement verifies the Text property for the buttonConnect button in the WysiCorp CRMLogin window is Connect. If the button text is incorrect, the checkpoint fails and an error message is displayed in the Errors pane.

```
Window("WysiCorp CRMLogin").Button("buttonConnect").Checkpoint("Text", "Connect", True, "The Connect button text is incorrect")
```

Verifying focus on a control

In this example, a checkpoint verifies a tab has focus or is currently selected. The Checkpoint statement verifies the Focused property for the Accounts tab on the tabControl tab bar in the WysiCorp CRM window is True. If the tab does not have focus, the checkpoint fails and an error message is displayed in the Errors pane.

```
Window("WysiCorp CRM").TabBar("tabControl").Tab("Accounts").  
Checkpoint("Focused", True, True, "The Accounts tab does not have focus")
```

Verifying items in lists

In this example, a checkpoint verifies a combo box contains the correct values. The Checkpoint statement verifies the ListItems property for the projectNameComboBox control in the WysiCorp CRMLogin window contains the following items: Seapine Software, WysiCorp, and ACME Bank. If this list does not contain these items, the checkpoint fails and an error message is displayed in the Errors pane.

```
Window("WysiCorp CRMLogin").ComboBox("projectNameComboBox").  
Checkpoint("ListItems", "Seapine Software,WysiCorp,ACME Bank", True, "The  
Projects list does not contain the correct values")
```

Using Window and Control Checkpoints Scenario

This scenario describes how to create window and control checkpoints that help you verify your application is working as expected. Window and control checkpoints can be created during recording or added to an existing script.

Setup

Before you begin this testing scenario, take the time to perform the following setup tasks:

- **Set up the testing environment**—Make sure you have access to the hardware, software, and network resources you need to record or run tests.
- **Create a workspace**—Workspaces organize scripts and related reports and datasheets. Depending on your organization's process, you may use one workspace for each application or one workspace for each functional area.
- **Set up the application in the application repository**—Application repositories store information about the tested application and version. Each version contains window and control data that identifies and locates objects.
- **Set general and playback options**—Take a few minutes to set general options that control how QA Wizard Pro works and set playback options that control how scripts run.
- **Record or update scripts**—Make sure you have recorded or updated all the scripts you want to edit to include checkpoints.

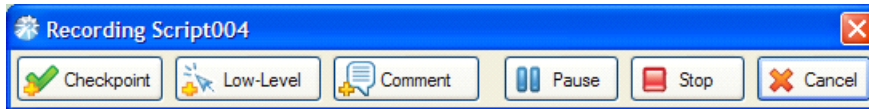
Using Window and Control Checkpoints Instructions

You can create window and control checkpoints during recording or after you record a script. It is generally more efficient to create checkpoints while recording a script.

Creating record-time checkpoints

While you are recording a new script, you can click the Checkpoint button on the Recording toolbar to quickly create a checkpoint.

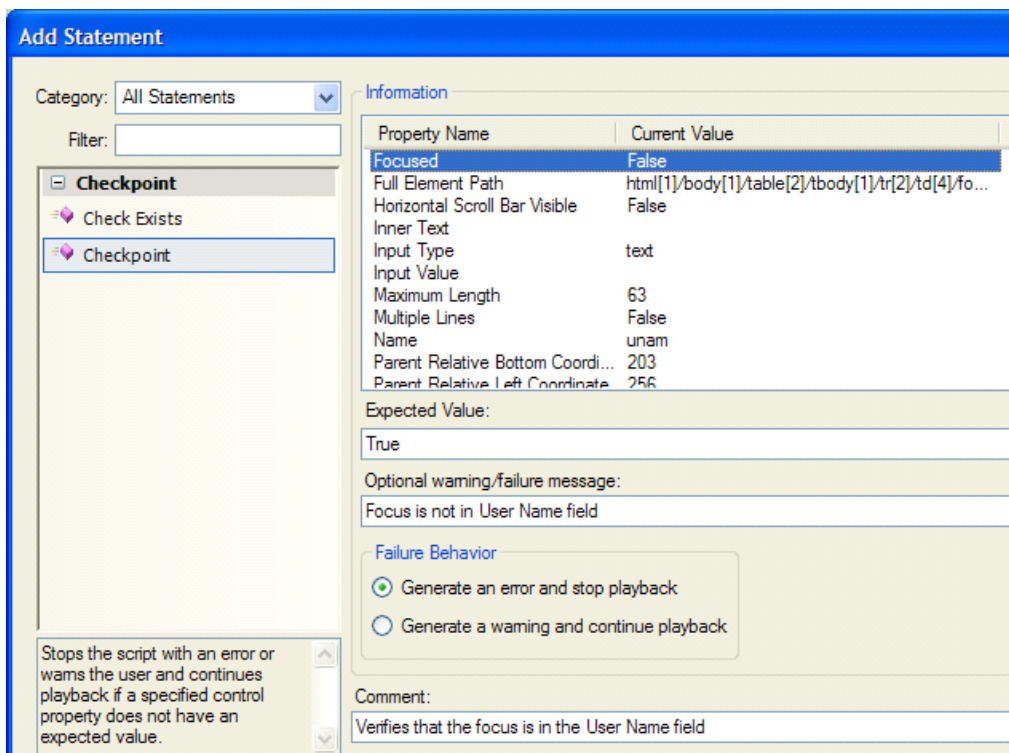
1. Click the **Checkpoint** Recording toolbar button while you are recording a script.



QA Wizard Pro switches to checkpoints mode. In this mode, the selected window or control is outlined with a red box.

2. Click the window or control you want to create a checkpoint for.

The Add Statement dialog box opens with the Checkpoint statement selected.



Tip: Right-click a window or control to display a menu of available controls to check. As you mouse over a control, it is outlined with a red box. Click the control in the menu to create the checkpoint on it. This method helps when you want to check items in tables or nested controls that are not visible or easy to find.

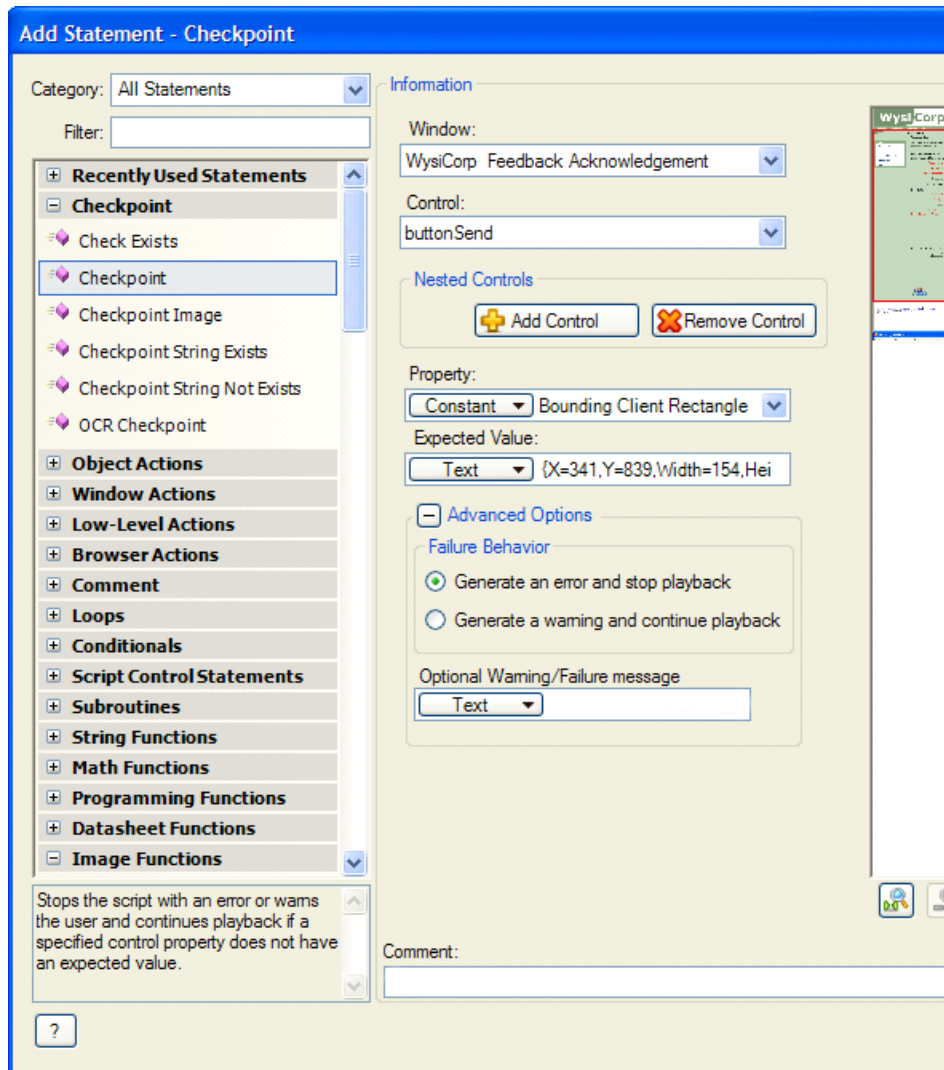
3. Select the **Property** to check from the list. The **Expected Value** field is automatically populated with the value in the Current Value column.
4. Enter an **Optional warning/failure message** to display in the Errors pane and add to the run report if the checkpoint fails.
5. Select a **Failure Behavior** option.

- Select **Generate an error and stop playback** to stop script playback, display a message in the Errors pane, and add the message to the run report.
 - Select **Generate a warning and continue playback** to continue running the script, display a message in the Errors pane, and add the message to the run report.
6. Optionally enter a **Comment**.
 7. Click **OK** to add the checkpoint to the script.
 8. When you are finished, click the **Checkpoint** Recording toolbar button to exit checkpoints mode and finish recording or add other checkpoints.

Creating checkpoints in existing scripts

You can create checkpoints in existing scripts to check any window or control property and verify it is working as expected. The available properties vary based on the type of control.

1. Open the script you want to add a checkpoint to.
2. Select the step that interacts with the control you want to check or the step above where you want to add the checkpoint.
3. Choose **Script > Add Statement**.
The Add Statement dialog box opens.
4. Select **Checkpoint** in the Checkpoint category.



5. Select the window to check from the **Window** list.
A screenshot of the selected window is displayed to the right of the Window list.
6. If you are creating a control checkpoint, select the control to check from the **Control** list.
The selected control is outlined with a red box in the screenshot.
7. Select **Item within** to check items in a combo box, menu, list box, tab control, or tree.
8. Select the **Property** to check from the list. The available properties change based on the selected window or control.
9. Enter or select the **Expected Value** to compare against the actual value.
10. Click **Advanced Options** to specify what happens if the checkpoint fails.
11. Select a **Failure Behavior** option.
 - Select **Generate an error and stop playback** to stop script playback, display a message in the Errors pane, and add the message to the run report.

- Select **Generate a warning and continue playback** to continue running the script, display a message in the Errors pane, and add the message to the run report.
12. Enter an **Optional Warning/Failure Message** to display in the Errors pane and add to the run report when the checkpoint fails.
 13. Optionally enter a **Comment**, such as the purpose of the checkpoint.
 14. Click **OK** to add the Checkpoint statement to the script.

Using Window and Control Checkpoints Conclusions

By using window and control checkpoints in scripts during recording or after, you can verify that your application is working as intended. This includes verifying window position, focus, specific text, and the number of items displayed in a list.

Links to other resources

You can view the following resources for more information on QA Wizard Pro.

- [QA Wizard Pro Resource Center](http://www.seapine.com/qawealtools.php)—www.seapine.com/qawealtools.php
- [QA Wizard Pro Blog](http://blogs.seapine.com/category/products/qawizardpro/)—http://blogs.seapine.com/category/products/qawizardpro/
- [Knowledgebase](http://www.seapine.com/kb/categories/QA+Wizard+Pro/)—www.seapine.com/kb/categories/QA+Wizard+Pro/