

About Using Image and OCR Checkpoints

You can create image and optical character recognition (OCR) checkpoints in QA Wizard Pro that help verify images and images with text, such as a Submit button, are in the correct location and meet expected values. This How To walks you through creating image and OCR checkpoints in scripts.

Using Image and OCR Checkpoints Overview

An image checkpoint fails when an image you are verifying does not match the accuracy or location expected during playback. An OCR checkpoint fails if the image does not contain the expected text during playback. You can configure the failure behavior to stop or continue playback and display an error or warning in the Errors window.

The chart below explains image and OCR checkpoints in more detail:

Type	Description
Image	Verify an image in the application against an expected image. For example, you can verify an image contains the correct colors or is displayed in the correct location.
OCR	Verify graphical text and text in objects that cannot be recognized in window and control checkpoints. Optical character recognition (OCR) checkpoints can be used to verify text in web pages, images such as chart and graphs, and Flash-based applications.

OCR checkpoint example

In this example, an OCR checkpoint is used to verify text in web page images. The ACME Bank login page menu bar uses images as buttons. The text in each image needs to be checked when the login test script runs. The first menu bar button is PRODUCTS, which is used as the expected text in the OCR checkpoint. The accuracy is set to 92%. When the script runs, if the expected text and selected area match with 92% or more accuracy, the checkpoint passes. If the accuracy is 91% or less, the checkpoint fails and an error message is displayed in the Errors window.

```
CheckpointOCR(OCRWindowRelative, 92, "PRODUCTS", 50, 163, 33, 32, True, "Products text is incorrect")
```

Note: The example shows the statement as you would enter it in Text View. You can also use the Add Statement dialog box to add statements in Grid View.

Using Image and OCR Checkpoints Scenario

This scenario describes how to create image and OCR checkpoints that help you verify images meet expected criteria and the text contained in web pages or images is correct. Image and OCR checkpoints can only be created in 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 Image and OCR Checkpoints Instructions

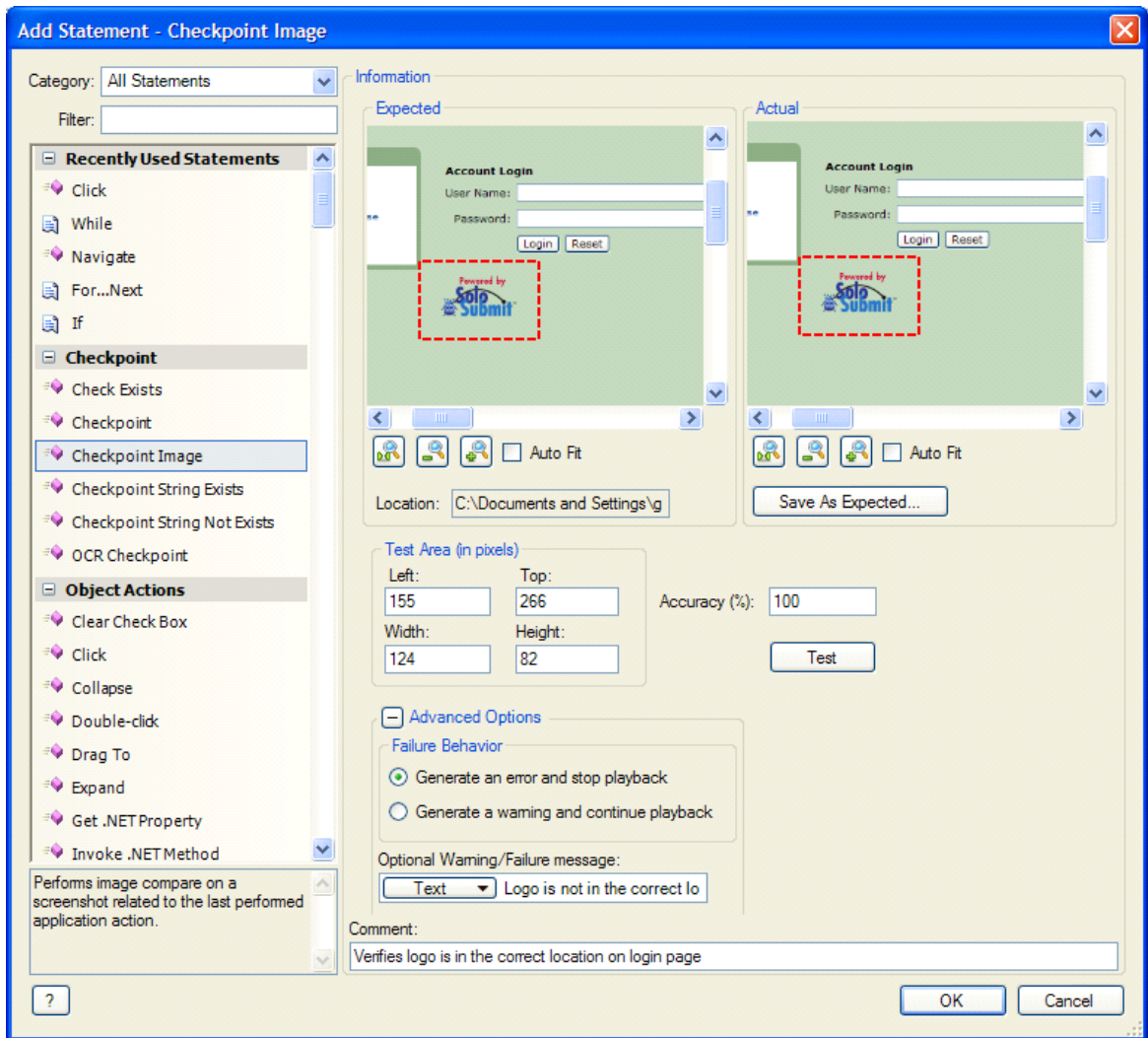
Image and OCR checkpoints can be used to assist in regression testing your application by verifying that images and graphical text or graphs meet expected criteria and are in the correct location.

Note: Image and OCR checkpoints can only be added to existing scripts.

Creating image checkpoints

Image checkpoints verify that an image in an application matches an expected image. For example, you can use image checkpoints to verify a company logo on a web page is in the correct location.

1. Select the step that is above where you want to add the image checkpoint. You must select a step that performs an action in the application.
2. Choose **Script > Add Statement**.
The Add Statement dialog box opens.



3. Select **Checkpoint Image** in the Checkpoints category.
4. Drag the red box on the **Actual** screenshot (right screenshot) to select the area you want to check. You can resize the box to fit the area.

Note: If the screenshot does not include the image you want to check, run the script before adding the checkpoint.

5. Enter the **Accuracy** of the match between the actual image and the expected image.
This is the minimum percentage required to pass the checkpoint. The actual image and expected image may not be exact matches, but if the accuracy is high enough it passes. If the accuracy is calculated as lower than this value, the checkpoint fails.
6. Enter the number of pixels to set the location and size of the **Text Area** to check.
These fields are automatically populated when you select an area to check, but you can change the values to adjust the location of the selected area.

- **Left** is the location of the left side of the red box that selects the area to check.
 - **Top** is the location of the top side of the red box that selects the area to check.
 - **Width** is the width of the red box in the screenshot that selects the area to check.
 - **Height** is the height of the red box in the screenshot that selects the area to check.
7. Click **Save As Expected** to save the actual image as the expected image used to compare with the application during playback.
The Save Image File As dialog box opens.
 8. Select the file location, enter a file name, and click **Save**.
The expected file is saved.
 9. Click **Advanced Options** to specify the **Failure Behavior** option for the checkpoint.
 - 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.
 10. Enter an **Optional Failure/Warning Message** to display in the Errors pane and add to the run report when the checkpoint fails.
 11. Optionally enter a **Comment**.
 12. Click **OK** to add the CheckpointImage statement to the script.

Creating OCR checkpoints

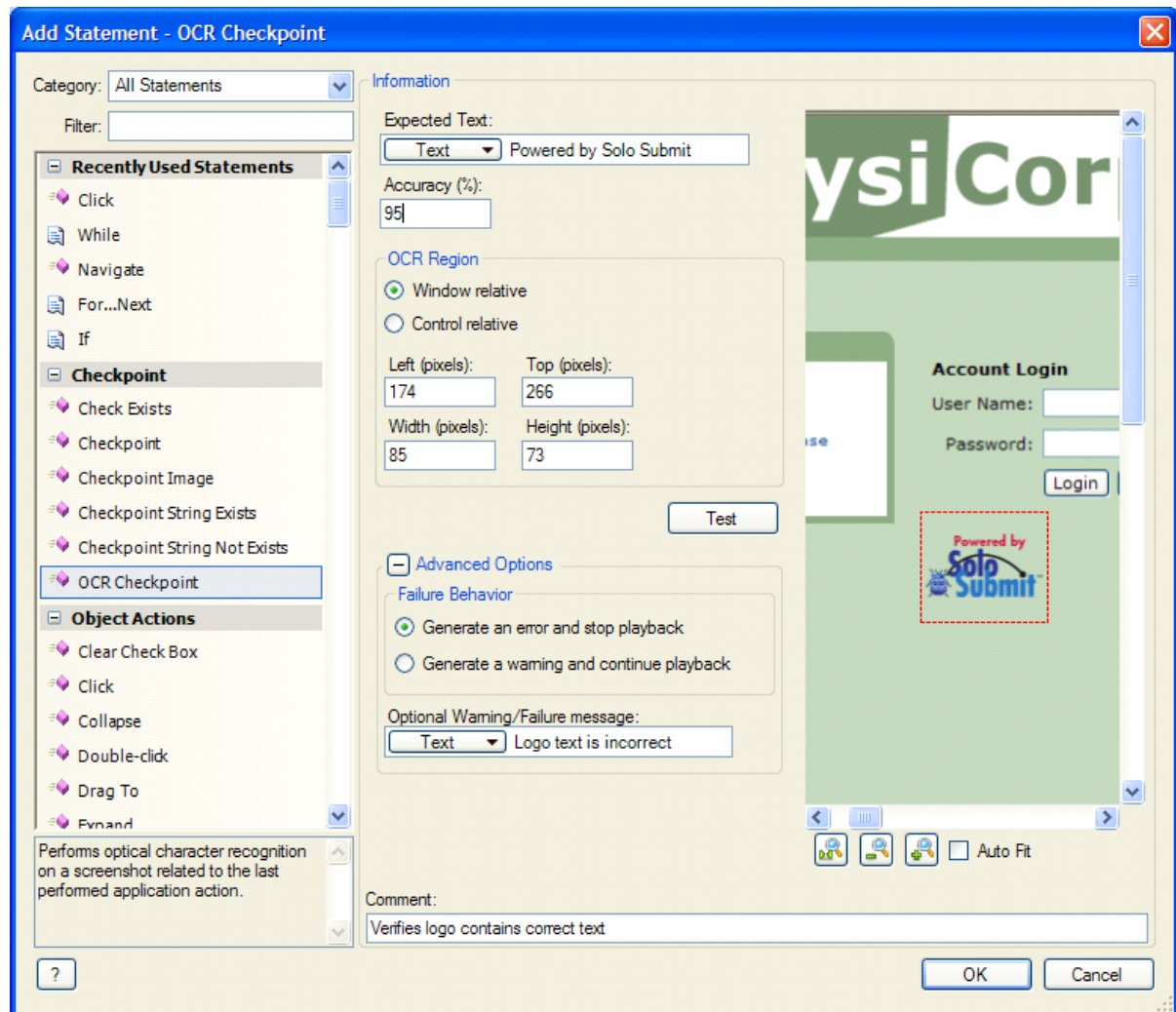
QA Wizard Pro can use optical character recognition (OCR) technology to read and convert graphical representations of characters into text. OCR checkpoints allow you to verify text in objects that QA Wizard Pro cannot recognize using window or control checkpoints.

For example, you can use OCR checkpoints to verify text in chart and graph images, or in Flash-based applications.

Keep the following in mind when creating OCR checkpoints:

- Run scripts at least once before creating an OCR checkpoint. The screenshot images used to create OCR checkpoints are generated during playback.
 - OCR recognizes a wide variety of fonts but may not accurately read all fonts.
 - OCR best recognizes dark-colored text on a contrasting background (e.g., black text on a white background). OCR cannot recognize white text and may not recognize light-colored text.
 - Use the highest screen resolution possible during recording when you plan to use OCR checkpoints in scripts. When you run the script, use the same resolution used during recording or the checkpoint may fail.
 - When OCR checkpoints are used in web scripts, do not resize the browser window during recording because this may cause the checkpoint to fail.
1. Select the step that is above where you want to add the OCR checkpoint. You must select a step that performs an action in the application.
 2. Choose **Script > Add Statement**.

The Add Statement dialog box opens.



3. Select **OCR Checkpoint** in the Checkpoints category.
4. Click the screenshot and select the area to check. The selected area is outlined with a red box.

Note: If the screenshot does not include the image you want to check, run the script before adding the checkpoint.

5. Enter the **Expected Text** to compare to the actual text. You can use a text string (default), expression, datasheet column value, variable, or a function.
6. Enter the **Accuracy** of the match between the actual text and the expected text.

This is the minimum percentage required to pass the checkpoint. The actual image and expected image may not be exact matches, but if the accuracy is high enough it passes. If the accuracy is calculated as lower than this value, the checkpoint fails.

7. Select how to locate the text in the tested application.

- **Window relative** locates text using the specified pixel locations, relative to the entire window.
 - **Control relative** locates text using the specified pixel locations, relative to the control in focus.
8. Enter the number of pixels to set the location and size of the area to check.

These fields are automatically populated when you select an area to check, but you can change the values to adjust the location of the selected area.

- **Left** is the location of the left side of the red box that selects the area to check.
 - **Top** is the location of the top side of the red box that selects the area to check.
 - **Width** is the width of the red box in the screenshot that selects the area to check.
 - **Height** is the height of the red box in the screenshot that selects the area to check
9. Click **Test** to make sure the checkpoint passes before adding it to the script.
- A dialog box opens that indicates if the expected text matches the actual text.
- Tip:** If the test fails, adjust the accuracy percentage until it passes.
10. Click **OK** to close the test dialog box.
11. Click **Advanced Options** to specify the **Failure Behavior** option for the checkpoint.
- 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 Failure/Warning Message** to display in the Errors pane and add to the run report when the checkpoint fails.
13. Optionally enter a **Comment**.
14. Click **OK** to add the CheckpointOCR statement to the script.

Using Image and OCR Checkpoints Conclusions

By using image and OCR checkpoints, you can verify an image is correct by comparing it against an expected image and verify graphical text or text in objects, such as charts and graphs. Image and OCR checkpoints also help you test updates made to a company logo, navigation buttons, and flash-based content.

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/