

Testing Dynamic Web Applications How To

You can use XML Path Language (XPath) queries and URL format rules to test web sites or applications that contain dynamic content that changes on a regular basis. XPath queries are used to find elements in HTML documents and URL format rules are used to test web applications with dynamic URLs. This How To walks you through configuring testing scripts to use XPath queries and URL format rules.

Testing Dynamic Web Applications Overview

An application or web site that contains dynamic content can be difficult to test. As the content changes, test scripts may fail because the application no longer matches the data stored in QA Wizard Pro. You can use XPath queries and URL format rules to configure your scripts to handle dynamic data.

About XPath queries

XPath queries find elements in HTML documents by assigning each element a unique path based on the hierarchy of all elements in the document. XPath is similar to a file path in an operating system. QA Wizard Pro uses XPath queries to easily locate controls when running scripts against dynamic web data.

XPath queries allow QA Wizard Pro to locate and perform actions on HTML elements that are dynamic or are not stored in the application repository. For example, you can use XPath queries to:

- Specify the location of an HTML control stored in the application repository. See [Using the XPath search method, page 2](#)
- Perform a checkpoint, object action, or low-level action on an HTML element that is not stored in the repository. See [Using XPath queries in checkpoints and actions, page 4](#).

Note: XPath language is defined by the W3C and is available online at www.w3.org/TR/1999/PR-xpath-19991008.html.

About URL format rules

If you test a web application with dynamic URLs, you need to modify the application repository so QA Wizard Pro can locate controls during playback. If a control property, such as Href or URL, is selected as search criteria and the expected value contains a URL, QA Wizard Pro cannot locate the control when the URL changes because the expected value and actual value of the URLs are different. Dynamic URLs may also cause QA Wizard Pro to add the same web page and controls to the application repository multiple times, which makes it unmanageable over time.

You can create URL format rules to compare, ignore, or perform regular expression searches on individual parts of URLs, such as the domain and parameters. For example, if a URL has a session ID parameter, you can create a URL format rule that ignores it. When you run the script, QA Wizard Pro ignores the session ID and searches for controls based on the rest of the URL. If you are testing a web application hosted on different web servers, you can create a URL format rule that compares the domain name URL to multiple domain names specified in a regular expression.

URL format rules can also be automatically applied to window and control properties captured when recording scripts or populating the repository. For example, if a URL format rule ignores the session ID and formats are automatically applied during recording, URLs for control properties added to the repository do not include the session ID.

QA Wizard Pro allows you to:

- Create URL format rules to specify how to handle dynamic URLs in web applications. See [Creating URL format rules, page 5](#).
- Apply the rules globally to all URLs in control properties for an application version, including existing controls in the repository. See [Applying URL format rules, page 7](#).
- Modify control properties individually to specify URL format rules to apply during playback. See [Editing URL format rules for individual controls, page 8](#).

Testing Dynamic Web Applications Scenario

This dynamic web applications scenario describes how to configure your scripts to use XPath queries or URL format rules to test a web application that contains dynamic data.

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 used to test the dynamic web application.

Testing Dynamic Web Applications Instructions

You can use XPath queries and URL format rules to test dynamic web applications using QA Wizard Pro. XPath queries are used to find elements in HTML documents that you want to perform an action on. URL format rules allow you to configure scripts to recognize dynamic URLs.

Using the XPath search method

QA Wizard Pro includes an XPath search method that locates the HTML controls in an application that is stored in the application repository. When you record a web script, the Standard search method is selected by default for most controls.

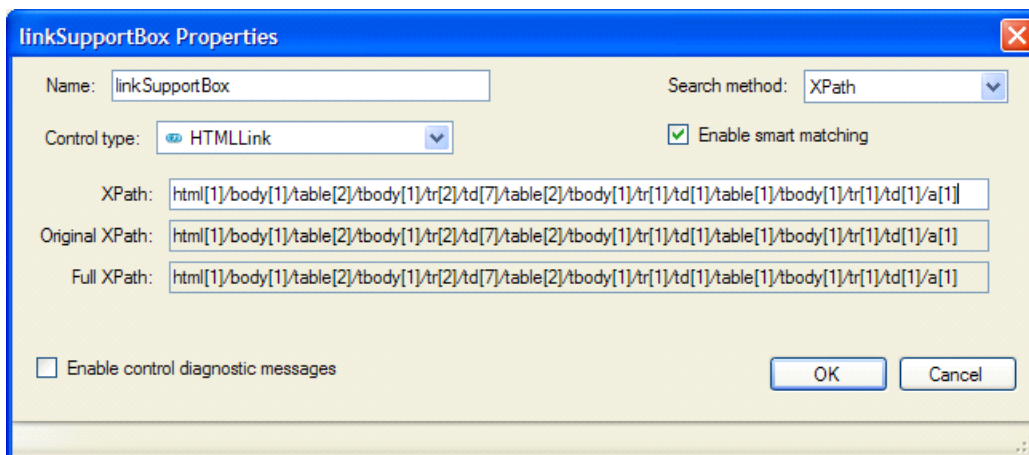
Note: Some controls use the XPath search method by default and the method cannot be changed.

The XPath search method uses an XPath query to search for an HTML control instead of the control properties in the Standard search method. Each control in the application repository has a unique XPath, which is captured when you populate the repository. When the script runs, the XPath query is performed and QA Wizard Pro interacts with the first HTML element that matches the query.

For example, a web site displays a list of products that all have an Add to Cart button for every item. QA Wizard Pro cannot distinguish between the Add to Cart buttons because each one has the same properties. You can use the XPath search method and specify an XPath query to locate a specific button based on its location in the hierarchy of elements on the page.

1. Right-click a control in the Application Repository pane and choose **Properties**.

The Properties dialog box opens.



2. Select **XPath** from the Search method list.

XPath information is displayed in the following fields:

- **XPath** displays the XPath query used to search for the control. The value in the Original XPath field is used by default.
- **Original XPath** displays the original XPath captured during recording. If the page has ID attributes, the XPath is truncated and used the ID to locate the control.
- **Full XPath** displays the full hierarchical XPath captured during recording.

3. Optionally modify the **XPath** query used to find the control during playback.

The Original XPath captured during recording is used by default, but you can modify it if the application changes.

Tip: If two or more controls have the same XPath ID, QA Wizard Pro will not find the correct control to perform the action on. You can copy the Full XPath value and paste it in the XPath field to locate the control using the full XPath.

4. Click **OK** to save the changes.

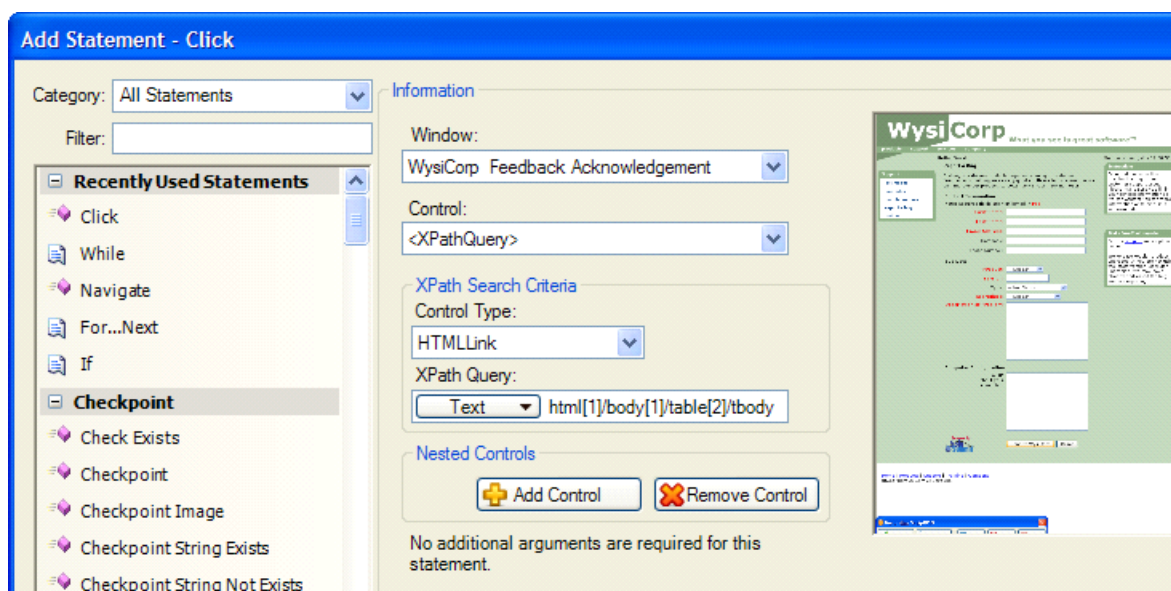
Using XPath queries in checkpoints and actions

Dynamic elements of an application may not be stored in the application repository because actions cannot be performed on them. You can use an XPath query when the element you want to perform an action on is not stored in the repository as a control. For example, you can use an XPath query in a checkpoint to verify an image path for an SRC attribute on a web page matched an expected value.

You can also use XPath queries with object actions, such as Click, or low-level actions, such as Mouse Move.

1. Open the script you want to modify to use an XPath query.
2. Choose **Script > Add Statement**.

The Add Statement dialog box opens.



3. Select the statement you want to add from the list of statements. To narrow the results, you can:
 - Select a category from the **Category** list.
 - Enter a **Filter**. For example, enter `Click` to exclude all statements except click-related statements.
4. Select the window that contains the element from the **Window** list.
5. Select **<XPath Query>** from the **Control** list.
6. If you are adding an object action or low-level action, select the **Control Type** to search for.
7. Enter the **XPath Query**. You can use a text string (default), expression, datasheet column value, variable, or a function.
8. Optionally, enter or select any additional information in the statement fields, and enter a **Comment**.
9. Click **OK**.

The statement is added to the script.

Creating URL format rules

You can use URL format rules to specify how to handle dynamic URLs. This allows QA Wizard Pro to locate a control when the URL changes. Format rules can be applied to URLs or only applied when URLs meet specific conditions.

Following is an example of a dynamic URL:

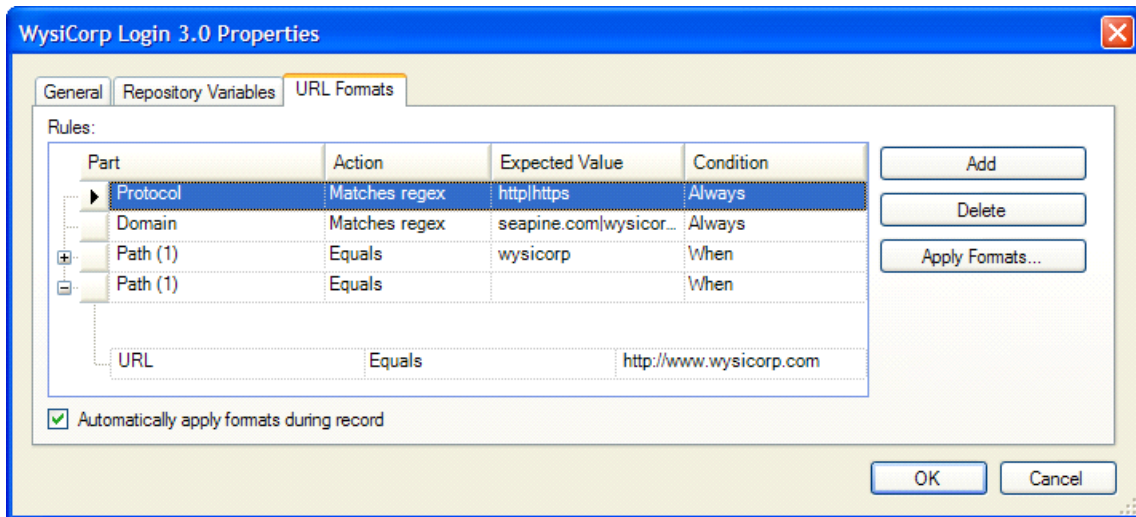
`http://www.wysicorp.com/wysicrm/news/2008/index.htm?date=20081015&lang=en#latest`

The parameters in the example URL are likely to change after you record the script, which may cause the script to fail if control properties used as search criteria use the URL as the expected value. URL format rules allow QA Wizard Pro to ignore specific parameters when running a script.

You can create URL format rules for the following parts of a URL:

Part	Description	Example
Protocol	Protocol used to retrieve the requested page from the web server	http
Domain	Domain where the web server is hosted	www.wysicorp.com
Path	Location on the web server	wysicrm news 2008
File	File requested from the web server	index.htm
Parameters	Name/value pair appended to a URL that passes information from the browser to the web server; parameters start with a question mark (?) and are separated by ampersands (&)	date=20081015 lang=en
Anchor	Named anchor, or bookmark, that identifies an internal section of a web page; anchors start with a pound sign (#)	latest

1. Select a web application version in the application repository.
2. Choose **File > Properties**.
The Properties dialog box opens.
3. Click the **URL Formats** tab.



4. Click **Add**.

An empty row is added to the Rules list.

5. In the new row, select or enter a **Part**.

Note: If the URL path has multiple levels, you can specify each path value separately. For example, if the path is three levels deep, add Path(1), Path(2), and Path(3) parts. Any values entered except Protocol, Domain, Path, File, or Anchor are interpreted as URL parameters.

6. Select an **Action**.

- **Equals** searches for exact matches to the expected value. This is the default action.
- **Starts with** searches for values that begin with the expected value.
- **Ends with** searches for values that end with the expected value.
- **Contains** searches for values that include the expected value.
- **Matches regex** searches for values in the URL part based on a regular expression. QA Wizard Pro supports .NET regular expressions.
- **Exists** verifies the expected value exists. This action is only available for anchors and parameters.
- **Ignore** ignores the part. This action is only available for anchors and parameters.

7. Enter the **Expected Value**.

QA Wizard Pro searches for the expected value in the URL and applies the rule if it is found. An expected value is not required if Exists or Ignore is selected in the Action column.

8. Select a **Condition** to indicate when to apply the rule to URLs.

- **Always** applies the rule to all URLs. This is the default condition.
- **When** applies the rule to URLs based on a specified condition. If you select When, a nested row is added under the current row. Select an Action and enter the URL to apply the rule to.

9. Optionally click **Apply Formats** to update URLs in the repository.

- Optionally select **Automatically apply formats during record** to apply the rules during future recording sessions.

Note: If this option is selected, the action of every control property with a URL value in the application version is set to Matches format. The format rules are applied when controls are added to the repository after recording or populating.

- Click **OK** to save the changes.

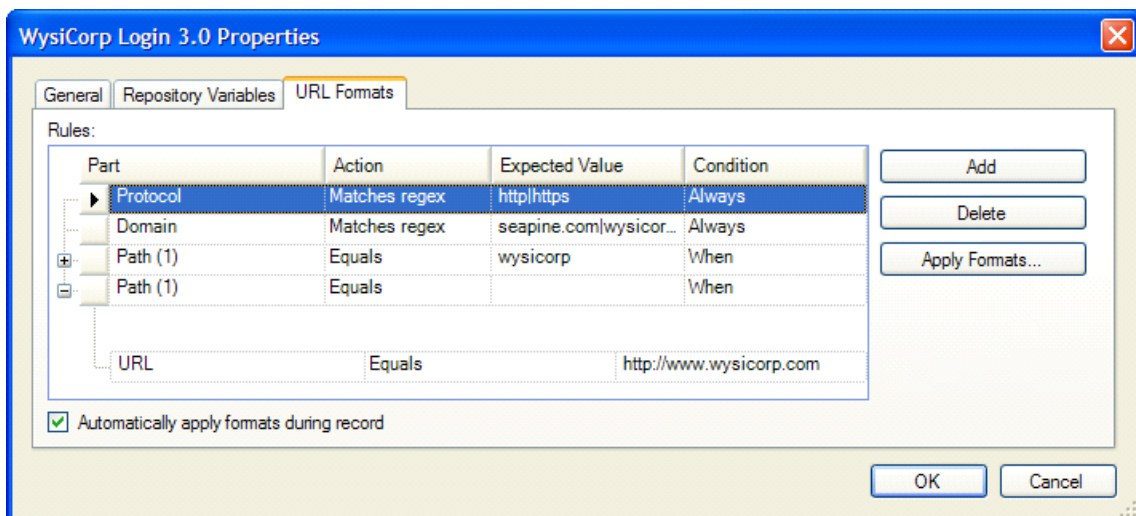
Applying URL format rules

You can apply URL format rules to all existing windows and controls for a web application in the application repository. Applying URL formats compares the format rules to all control properties with URL expected values and updates each URL if the conditions apply.

- Select a web application version in the application repository.
- Choose **File > Properties**.

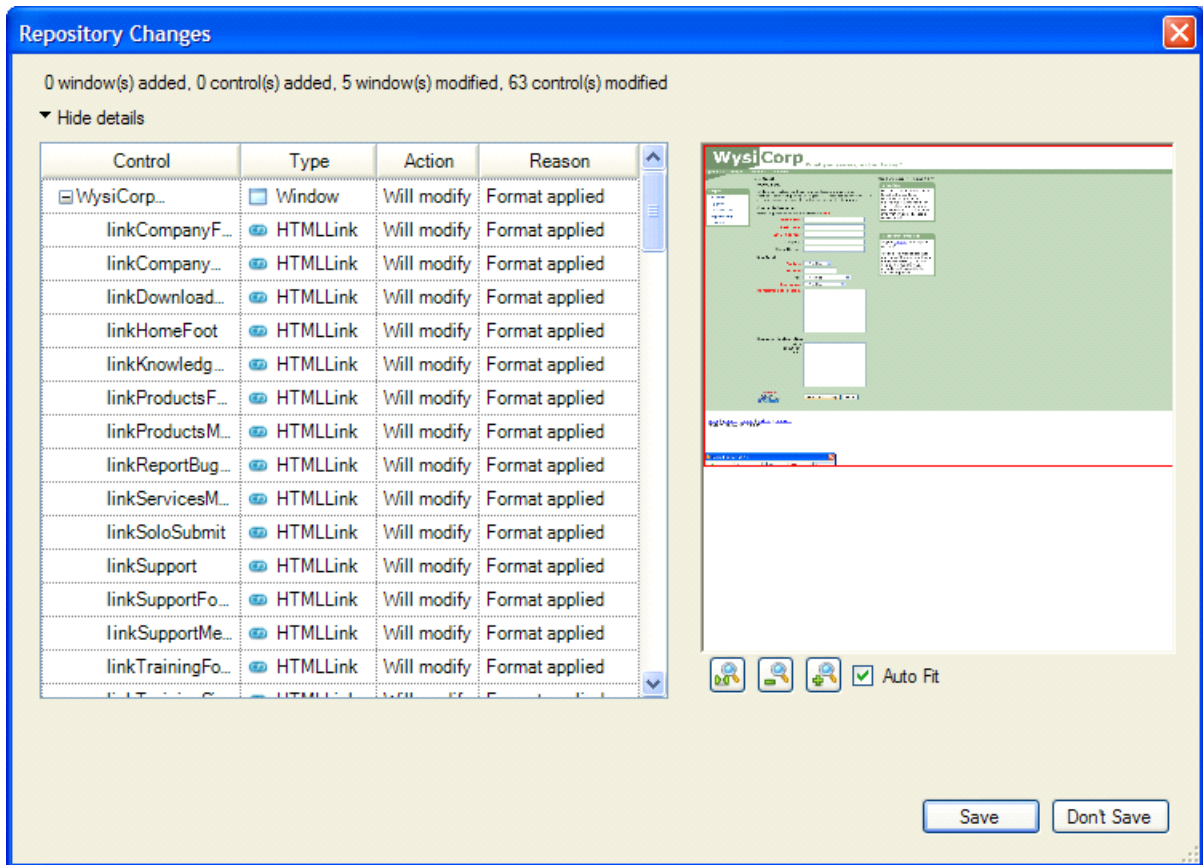
The Properties dialog box opens.

- Click the **URL Formats** tab.



- Click **Apply Formats**.

The Repository Changes dialog box opens and displays windows and controls that contain URL format changes.



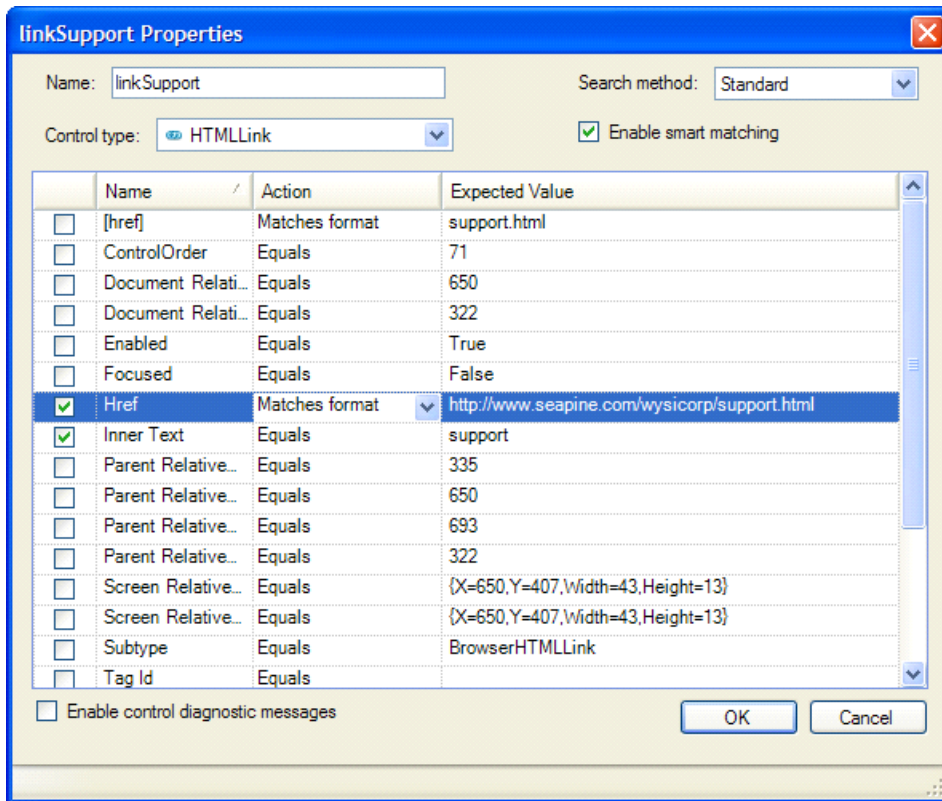
5. Click **Save** to save the changes and update the repository.
6. Click **OK** to close the Properties dialog box.

Editing URL format rules for individual controls

If you only need to change the URL format for a few controls in the repository, you can modify each control individually and set URL expected values, such as Href, URL, and SRC, to match a specified format during playback.

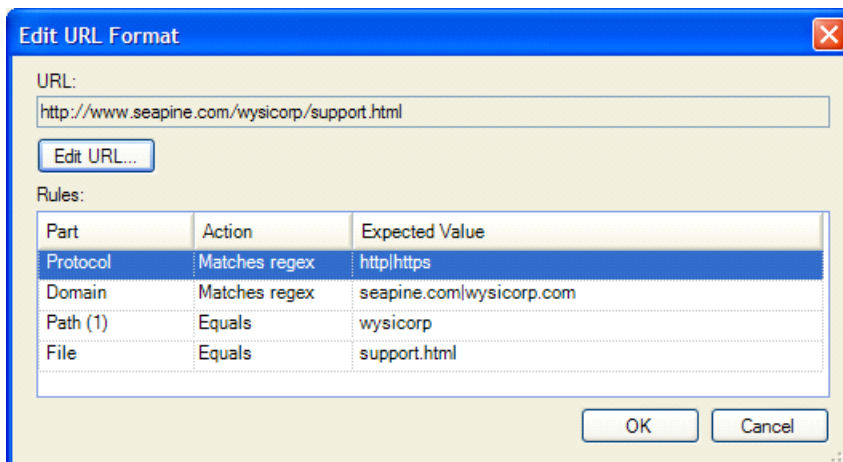
1. Locate the control you want to edit the URL format rules for in the Application Repository pane.
2. Right-click the control and choose **Properties**.

The Properties dialog box opens.



3. Select the row for the control with the URL in the expected value.
4. Select **Matches format** in the Action column.
5. Click **Edit** in the Expected Value column.

The Edit URL Format dialog box opens.



The following information is displayed for each part of the URL:

- **Part**—Name of the URL part.
- **Action**—How QA Wizard Pro searches for the expected value in the URL.
- **Expected Value**—Value QA Wizard Pro searches for in the URL based on the selected action.

6. Click **Edit URL** if you need to change the URL.
7. Optionally change the **Action**.
 - **Equals** searches for exact matches to the expected value. This is the default action.
 - **Starts with** searches for values that begin with the expected value.
 - **Ends with** searches for values that end with the expected value.
 - **Contains** searches for values that include the expected value.
 - **Matches regex** searches for values in the URL part based on a regular expression. QA Wizard Pro supports .NET regular expressions.
 - **Exists** verifies the expected value exists. This action is only available for anchors and parameters.
 - **Ignore** ignores the part. This action is only available for anchors and parameters.
8. Optionally change the **Expected Value**.
9. Click **OK** to save the changes.
10. Click **OK** to close the Edit URL dialog box.

Deleting URL format rules

You can delete URL format rules that are no longer used.

1. Select a web application version in the application repository.
2. Choose **File > Properties**.
3. Click the **URL Formats** tab.
4. Select the rule you want to delete.
5. Click **Delete**.

You are prompted to confirm the deletion.

6. Click **Yes**.

The rule is deleted.

Note: If a URL property, such as URL, Href, or Src is selected as search criteria and you delete a URL format rule, you need to manually update the search criteria if you no longer want the URL to match a specific format. In the window or control Properties dialog box, change the Matches format value in the Actions column to another action.

Testing Dynamic Web Applications Conclusions

Using XPath queries and URL format rules allows you to test web applications with content that changes frequently. You can modify all elements or individual controls in your scripts and have them automatically adjust to the dynamic data or URLs in your application. This helps you save time by keeping your scripts up-to-date and reducing the need to record new scripts every time content and URLs change.

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/