

Repository Variables How To

You can use repository variables to store values that can be used in all scripts recorded or run against an application version. This helps reduce the need for extensive script modifications and eliminates the maintaining of a large database of scripts. This How To walks you through creating, editing, and adding repository variables to your scripts.

Using Repository Variables Overview

Repository variables are similar to variables defined at the script level, but unlike script variables, repository variables are defined once and applied to the entire application. When you create a repository variable, a default value is identified, which is used every time a script is recorded or run against the application version.

For example, if you have a web application that is hosted on a development server, test server, and production server, you can create a repository variable that contains the values for each server address. You can also configure QA Wizard Pro to prompt you to select the variable value that corresponds to the server you are running or recording the script against.

About the URLROOT repository variable

When a web application is created in QA Wizard Pro, you can automatically create a URLROOT repository variable to test a web site hosted on multiple servers. The variable value is the server address that hosts the web site you are testing. For example, for the web site www.wysicorp.com/admin, the URLROOT variable value is www.wysicorp.com.

After the URLROOT variable is created, you can use it in any scripts. For example, you can add the variable before the RunApp statement to specify the address used to open the web site and replace other references to the variable throughout the script.

By default, the URLROOT variable has the following settings:

- Variable type is List
- Auto-record substitution is enabled
- Prompt is disabled

Tip: You can modify variable settings after you add the web application. See [Managing repository variable list values](#), page 4.

Using Repository Variables Scenario

This scenario describes how to create repository variables and use them in your scripts.

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 use with repository variables.

Using Repository Variables Instructions

After you create repository variables, you can assign them to your scripts as expressions, text strings, or values from an associated datasheet. If multiple scripts test the same data and the data changes based on application conditions or inputs, you can create repository variables to streamline scripts.

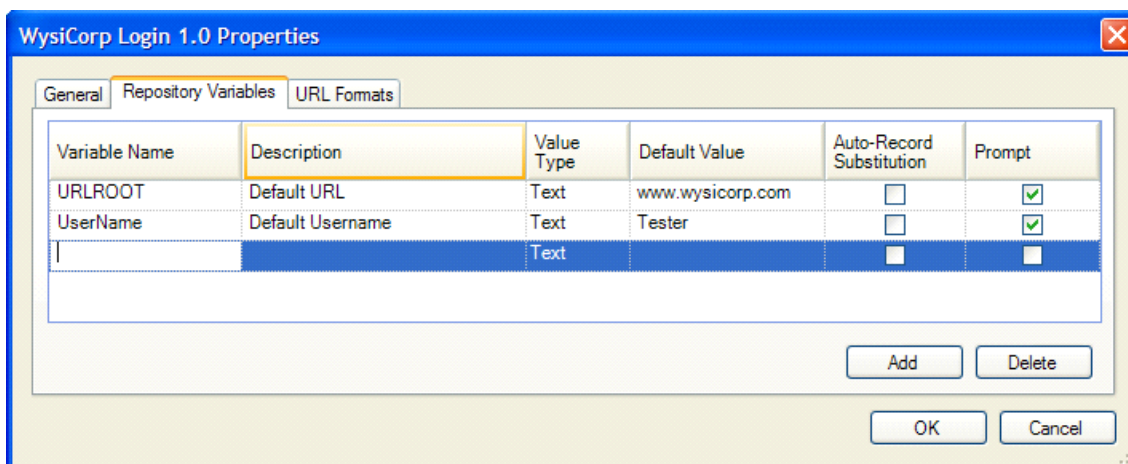
Creating repository variables

1. Select a version in the application repository and choose **File > Properties**.

The Properties dialog box opens.

2. Click the **Repository Variables** tab.
3. Click **Add**.

A new row is added to the variables list.



4. Enter a **Variable Name**.

Variable names must conform to the following rules:

- Must start with an alphabetic character
- Cannot contain spaces
- Can only contain alphanumeric characters or underscore characters

5. Enter a **Description**.
6. Click the arrow in the Value Type column and select a variable **Type**.
 - Select **Text** to assign one value to the variable.
 - Select **List** to assign multiple values to the variable. See [Managing repository variable list values, page 4](#).

7. Enter the **Default Value**. The value can be a text string, expression, value from an associated datasheet, a variable, or a function.

If you selected List as the Value Type, enter the list values separated with a comma or click the plus button to add the values. The first list value is the default value used when recording and running scripts.

8. Select **Auto-Record Substitution** to automatically replace all instances of the variable value with the variable name during recording.

The script and expected values for window and control properties are populated with the variable name instead of the variable value. This option is helpful if you run scripts against different web servers because you do not have to make extensive modifications to scripts and the application repository.

Note: Only select this option to replace unique values such as web server addresses. If the value is not unique, you may accidentally replace the wrong value in scripts and the application repository, which can cause scripts to fail.

9. Select **Prompt** to prompt users to enter a variable value.

Note: If this option is selected, users are prompted to enter a variable value when the RunApp or ConnectToProcess statements run. If the repository variable is set in the script before these statements, users are not prompted and the value set in the script is used.

10. Click **OK** to save the variables.

Using repository variables in scripts

Script variables can only be used in the script they are created for, but repository variables can be used in all scripts recorded for an application version. Repository variables only need to be defined once and are stored with a version in the repository.

For example, if you have a repository variable named USER_ID, to use it in a script, you enter `PrintLn (USER_ID)` in the script where you want to use the variable. During playback, the default value you entered for the variable is used.

Keep the following in mind when using repository variables in scripts:

- If a script variable and repository value has the same name, the script variable overrides the repository variable during playback.
- If you use a repository variable before the RunApp or ConnectToProcess statement, users are not prompted to select a variable value unless other repository variables require prompting.

- Repository variables are text values. To use variables in statements that use other value types, you must convert the values. For example, a variable named Time has a value of 30000. To use the variable in the SetControlSearchTimeout statement, which requires an integer value, you must use the following Integer function to convert the text value before the SetControlSearchTimeout statement is used: `var=Integer(Time)`.

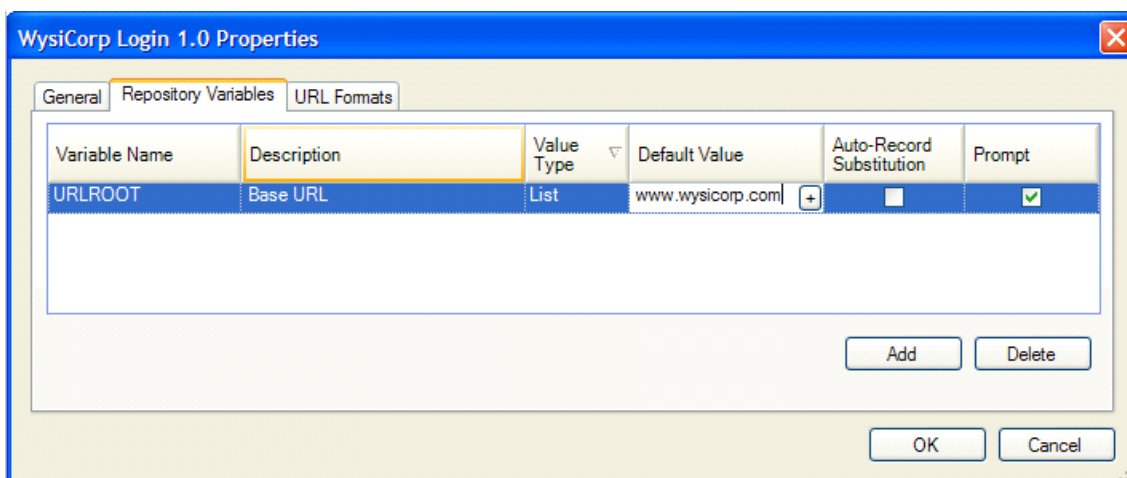
Managing repository variable list values

Repository variables can contain multiple list values that users can select from when they are prompted to enter a variable value.

1. Select a version in the application repository and choose **File > Properties**.

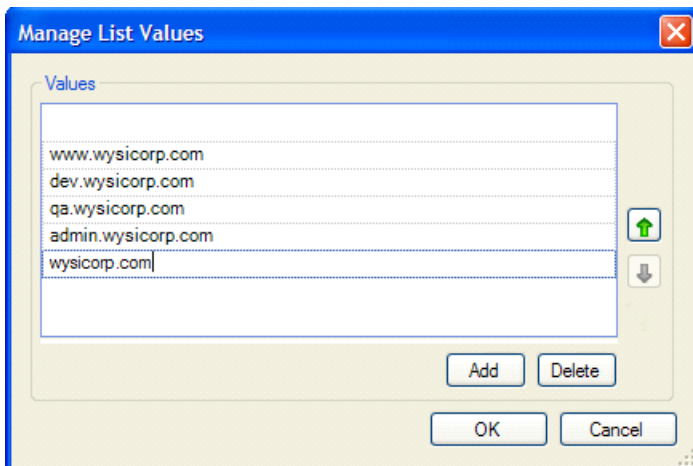
The Properties dialog box opens.

2. Click the **Repository Variables** tab.



3. Click the plus button in the Default Value column.

The Manage List Values dialog box opens.



4. Click **Add** to add a value.

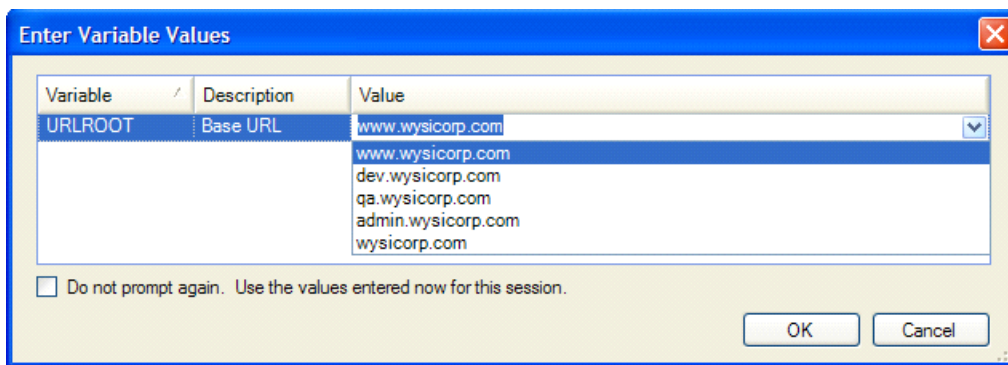
A new row is added

5. Enter the value in the new row.
Repeat steps 4 and 5 to add additional values.
6. Click a value to change it.
7. Select a value and click **Delete** to delete it.
You are prompted to confirm the deletion. Click **Yes** to confirm or **No** to cancel.
8. Select a value and click the arrow buttons to reorder the list.
The first value in the list is used as the default value when recording or running scripts.
9. Click **OK** to save the changes.

Entering repository variable values during recording and playback

Repository variables can be configured to prompt users to enter a variable value when recording or running scripts. Depending on the variable settings, you can enter a value or select it from a list.

1. If a repository variable is set to prompt, the Enter Variable Value dialog box opens when you record or run a script. The variables configured for the application version and their default values are displayed.



2. Optionally select a value or enter a new value in the **Value** column.

Note: If you enter a new value, it is saved and used as the default value the next time you record or run scripts against the application version.

3. Select **Do not prompt again** if you do not want the Enter Value Variables dialog box to open when you record or run subsequent scripts against the application version. The current value is used until QA Wizard Pro is restarted.
4. Click **OK**.
The value you selected or entered is used as the variable value.

Editing repository variables

You can edit repository variable names, values, and options.

1. Select a version in the application repository and choose **File > Properties**.
The Properties dialog box opens.
2. Click the **Repository Variables** tab.

3. Select a value and make any changes.
4. Click **OK** to save the changes.

Deleting repository variables

You can delete repository variables if they are no longer used.

1. Select a version in the application repository and choose **File > Properties**.

The Properties dialog box opens.

2. Click the **Repository Variables** tab.
3. Select the variable you want to delete.
4. Click **Delete**.

You are prompted to confirm the deletion.

Note: This action cannot be undone.

5. Click **Yes** to delete the variable.

Using Repository Variables Conclusions

You can create repository variables for scripts that are stored at the application version level. These variables can be used to replace usernames, passwords, URLs, and other dynamic data. They also help save time by reducing the amount of scripts you need to create or modify to thoroughly test your application.

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/