

Understanding QA Wizard Pro Repository Variables

Repository variables can help you test dynamic attributes of Windows and Web applications. Repository variables, which are defined in the application repository for each version, can be used in any scripts recorded or run against the application. During playback, any instance of a variable is replaced with the default value set for the variable or the value selected when playback starts. This article describes common uses for repository variables, where variables can be used, and how variables are used during recording and playback.

The primary benefit of using repository variables is that you can maintain reusable test scripts without spending extra time modifying or creating new scripts every time an application changes. Repository variables also allow you to manage and access dynamic test data without maintaining multiple data sources.

When to use repository variables

Use repository variables if you have multiple scripts that test the same data and the data changes based on application conditions and inputs. Dynamic data may include input values, URLs, window names, or application text. The following examples describe common uses for repository variables.

Web sites hosted on multiple servers

You may need to run the same scripts against multiple Web servers, such as development, test, and production servers. You can create a repository variable that contains each server address as a value and set the variable to prompt when recording or playback starts so users can select the address to use for the session. You can also use the variable in the URL to open the Web page.

When you create a new Web application, you can automatically create a URLROOT repository variable. The URLROOT repository variable stores Web server addresses so you can easily run scripts against multiple servers.

Applications with multiple login credentials

An application may include functionality that is only accessible when a specific username and password is used. Instead of maintaining multiple login scripts or modifying scripts to change static username and password values, you can create username and password variables. You can set the variables to prompt when recording or playback starts so users can select the appropriate login values.

Applications with dynamic controls

Dynamic applications may have controls with properties that change frequently, such as window names. QA Wizard Pro may use these properties to locate controls during playback. If the expected value of a property selected as search criteria does not match the application during playback, scripts fail. To avoid this, you can create a repository variable for the dynamic property and specify the variable as the expected value in the application repository.

Where to use repository variables

You can use repository variables in scripts, window and control expected values in the application repository, and Web application startup URLs.

In scripts

Repository variables are commonly used in scripts to replace static values in statements. For example, an application has a LOGIN_NAME repository variable and you want to replace any instance of a static username value in scripts with the variable name.

In the GUI statement editor, you can select Variable as the value type and select the variable. Figure 1 shows the LOGIN_NAME variable used in the Type Text statement.

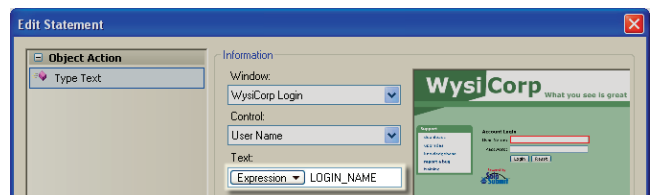


Figure 1: Using a repository variable in the GUI statement editor.

In Text View, you can replace the static value with the variable name. The following example shows the LOGIN_NAME variable used in the Type Text statement.

```
Window ("Login").Control("Username").TypeText(LOGIN_NAME)
```

You can also assign values to repository variables in scripts. In the GUI statement editor, use the Assignment statement (see Figure 2). In Text View, enter the variable name followed by an equals sign and the value, which can be text, an expression, or a value from an associated datasheet. For example:

```
LOGIN_NAME = "Developer"
```

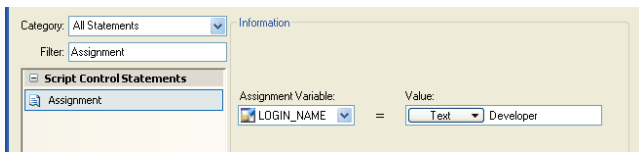


Figure 2: Assigning a repository variable value in the GUI statement editor.

You can use script variables to override repository variables during playback. If a script variable and repository variable have the same name, the script variable value overrides the repository variable value. After playback, the repository variable maintains its default value. You may want to establish a naming convention for repository variables to make sure duplicate names are not created.

Expected values for windows and controls

You can use repository variables in the expected values used as search criteria to locate windows and controls during playback.

For example, the URL for a hyperlink on a Web page changes based on the server address. If a property used as search criteria includes the URL, scripts will fail because the expected value does not match the Web page value. You can replace the static URL with a repository variable so QA Wizard Pro can successfully locate the hyperlink during playback. Figure 3 shows how the URLROOT repository variable is used instead of the static server address in the Href property.

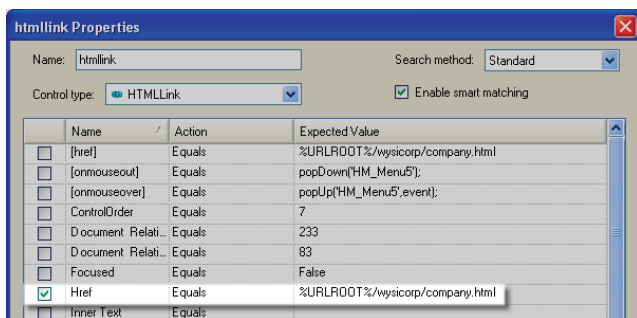


Figure 3: Using the URLROOT repository variable.

Web application startup URLs

If you add the URLROOT repository variable when you add a Web application, it is automatically used in place of the static URL in the application startup information. If you later decide to use a repository variable instead of a static URL, you can modify the URL in the application properties. Figure 4 shows how the URLROOT repository variable is used in a startup URL.

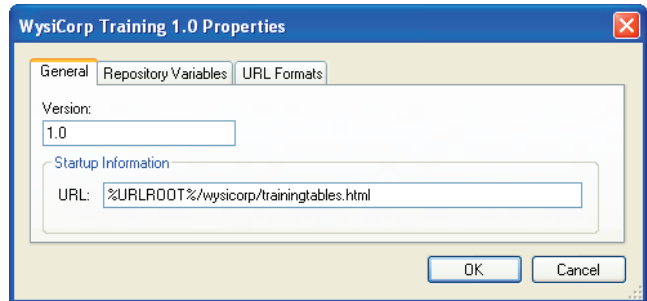


Figure 4: Using the URLROOT repository variable as the startup URL.

How repository variables are used during recording

Repository variables are only used during recording if the Auto-Record Substitution option is selected for the variable. Auto-record substitution automatically replaces any instance of the variable value in the script with the variable name. Any instances of the variable value in expected values for new controls captured during recording are also replaced with the variable name.

For example, if you use the URLROOT repository variable with a default value of http://www.seapine.com and the Auto-Record Substitution option is selected for the variable, all instances of http://www.seapine.com are replaced with %URLROOT% during recording.

You should only use auto-record substitution to replace unique values, such as Web server addresses. If the value is not unique, you may unintentionally replace the wrong value in a script and the repository, which can cause the script to fail during playback.

How repository variables are used during playback

During playback, the default repository value or a selected value is used for each instance of the variable name. For example, if the default value for the URLROOT variable is http://www.seapine.com, any instance of URLROOT in the script or repository is replaced with http://www.seapine.com.

Repository variables include a Prompt option, which provides flexibility to preset variable values when recording or running scripts. If the variable is set to prompt, the Enter Variable Values dialog box opens when recording or playback starts.

You can use the default variable value or enter a different value. If you use a different value, it is set as the new default variable value. Figure 5 shows three variables and their default values in the Enter Variable Values dialog box.

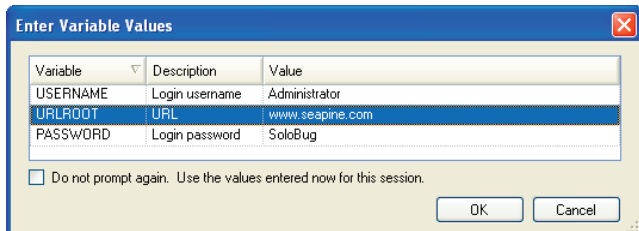


Figure 5: Using repository variables during playback.

Additional information

The QA Wizard Pro User Guide includes more information about repository variables. If you need to download a copy, visit www.seapine.com/documentation.html#QAW.