

Remote Testing How To

You can use the remote testing features of QA Wizard Pro to run your testing scripts from multiple computers or from computers in different locations. This How To walks you through planning, configuring, and running automated testing script from remote computers.

Remote Testing Overview

Remote testing allows you to use any number of remote computers to distribute your automated testing scripts. For example, you can run a script from a remote computer with a 64-bit operating system installed when you have a 32-bit operating system installed locally. This allows you to test your application with a much wider range of hardware/software configurations.

By using remote computers to distribute your load testing scripts, you can:

- Save time and complete your testing faster
- Test multiple operating systems
- Test multiple application builds (32-bit, 64-bit)
- Test various browsers such as Mozilla Firefox, Internet Explorer, or Google Chrome
- Test different computer configurations (processor type, amount of RAM, etc.)
- Test different application interfaces (command line, web browser, etc.)

Remote Testing Scenario

This remote testing scenario describes how to plan, create, configure, and run automated testing scripts from remote computers.

Setup

Before you begin this remote 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 that 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.

Planning questions

Ask yourself the following questions and plan your remote testing scripts around the answers:

- Are there 32-bit or 64-bit versions of the application? If so, what percentage of users are on the 32-bit version versus the 64-bit version?
- What versions of Windows are supported by the application?
- How does the user interact with the application? Is there also a web interface or command line?
- If the application is web-based, which web browsers are supported? Does this include the 64-bit versions of the browsers?
- Does the application require you to test many different computer configurations, such as multiple types of video cards or different amounts of RAM?

Note: The above questions are a good place to start, however you may end up with many more when researching and planning your remote testing scripts.

Running Scripts Remotely

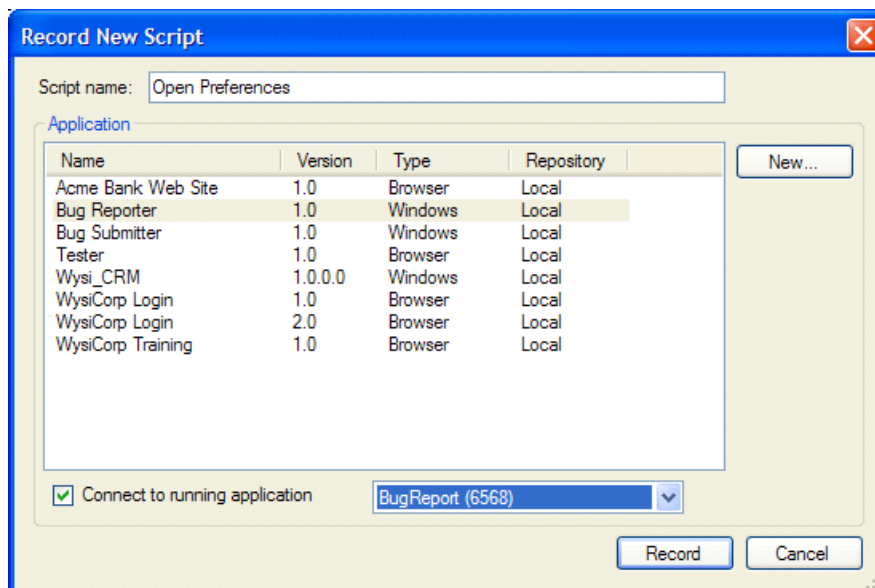
The test scripts you create capture the actions you perform as you interact with the application. After you finish recording the scripts, they can be edited to create more complex testing scenarios. The information gathered during the planning of your automated testing scripts and the answers to the planning questions (see [Remote Testing Scenario, page 1](#)) determine the complexity and amount of test scripts you create.

Note: Try to keep your testing scripts short and task-based. This makes them easier to manage and maintain. Tests should be repeatable and have an expected result.

Recording new scripts

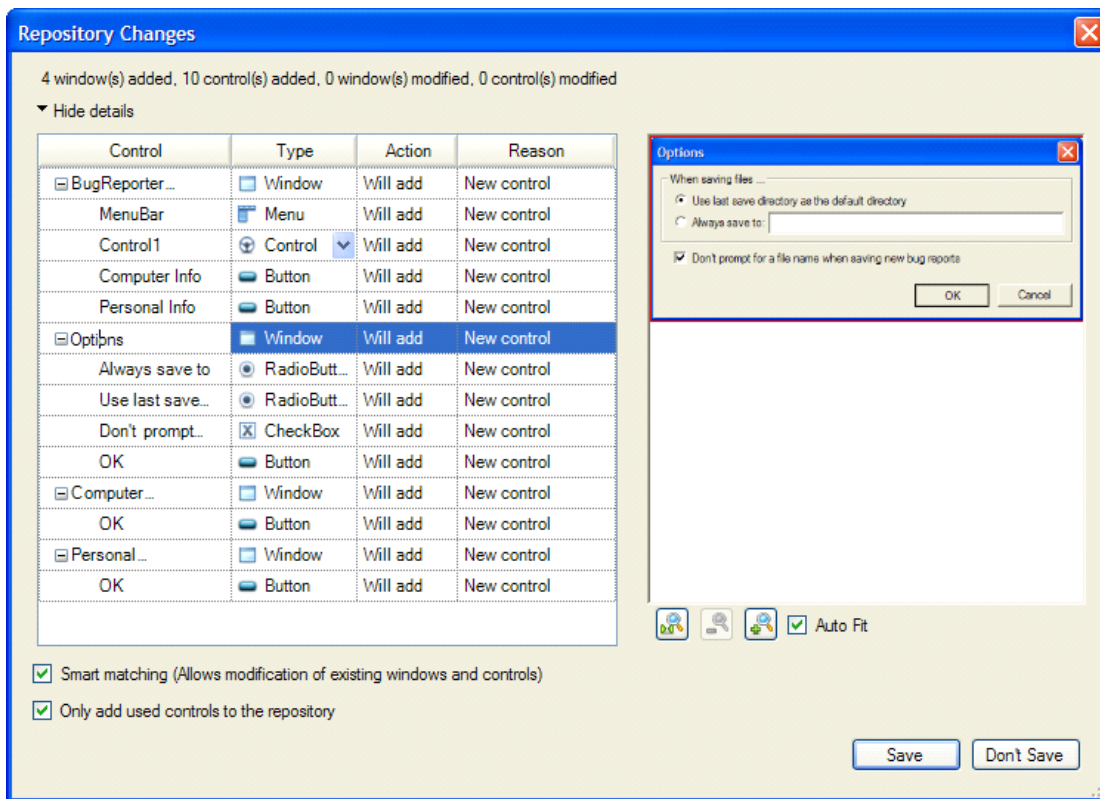
1. Choose **Script > Record New Script**.

The Record New Script dialog box opens.



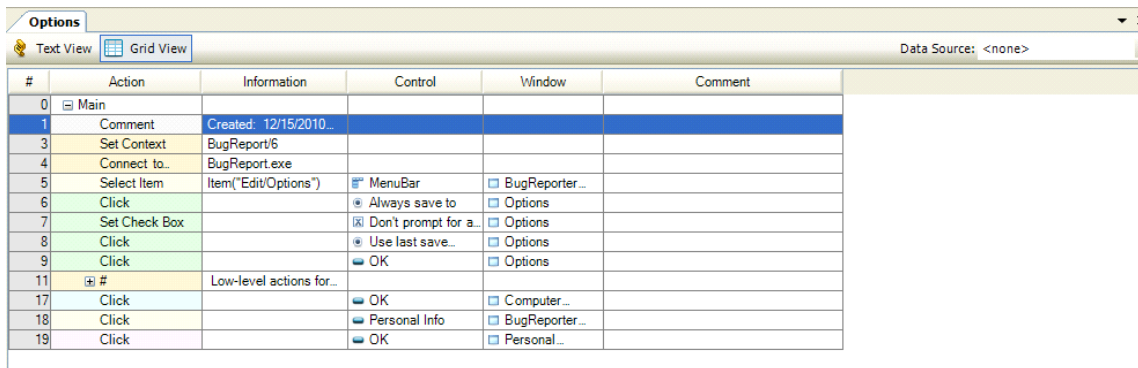
2. Enter a **Script name**.

3. Select the **Application** you want to record against.
Click **New** to add an application.
4. Select **Connect to running application** to record against a Windows application running on the test computer.
 - Select the application from the corresponding list. In order for the application to appear in the list it must be running on the test computer. If new windows and controls are found during recording, they are added to the repository. If you do not select the right application, windows and controls are added to the wrong place in the application repository.
 - If you record against a running application, the application must be running on the test computer during playback or the script fails.
5. Click **Record**.
The application starts and the Recording toolbar opens.
6. Navigate through the application, performing the actions you want to record.
 - Click the **Checkpoint** button on the Recording toolbar to add checkpoints to the script during recording. Click the button again to exit checkpoints mode.
 - Click the **Low-Level** button on the Recording toolbar to switch to low-level recording mode. Low-level action statements perform mouse actions that QA Wizard Pro records when it does not recognize an object. Click the button again to exit low-level mode.
 - Click the **Comment** button on the Recording toolbar to add a comment to the script. Comment statements are not run during playback.
 - Click the **Cancel Recording** button on the Recording toolbar to cancel recording and discard all recorded actions.
7. Click **Stop** on the Recording toolbar button.
The Repository Changes dialog box opens. New or changed windows and controls are displayed.



8. Click **Save** to save the repository changes.

The script is processed and the actions are added to it. Information about the application windows and controls are added to the repository.



Tip: Windows and controls added to the repository may have generic names depending on the application. You can modify the repository to make the names consistent with other objects.

9. Repeat steps 1 through 8 to create all the load test scripts necessary for testing your application.

Adding existing scripts

If you want to work with a script from another workspace, you can add an existing script. When you add an existing script, it is shared and not copied. Any changes you make to the script are made in all workspaces.

1. Choose **File > Add Existing Script**.

The Add Script dialog box opens.

2. Browse to the workspace folder, select the script, and click **Open**.

The script is added.

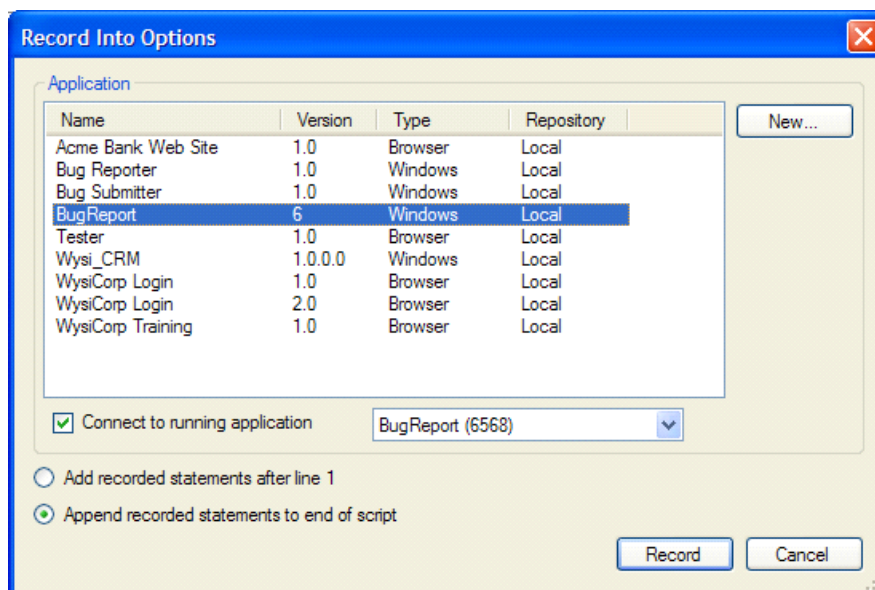
Tip: You can also use the add existing script command to add scripts that other users email to you. Save the files in a workspace folder. Choose File > Add Existing Script, select the script, and click Open.

Recording additional script steps

You can add additional steps to an existing script by recording them. New steps are added to the end of the script or to the line below a selected script step. You can also manually add steps, but recording the new steps ensures actions are performed on the correct windows and controls.

1. Open the script to add steps to.
2. Choose **Script > Record Into**.

The Record Into dialog box opens.



3. Select the **Application** you want to record against.

Click **New** to add an application.

4. Select **Connect to running application** to record against an instance of a Windows application running on the test computer.
5. Select a recording option.
 - Select **Add recorded statements after line** to add the new steps after the selected step.
 - Select **Append recorded statements to end of script** to add the new steps after the last step in the script.
6. Click **Record**.

You are prompted to run the current steps before recording.

Note: If repository variables are defined for the application version, you may be prompted to enter variable values before you begin recording.

7. Click **Yes**.

The script runs. The Recording toolbar opens when the script finishes running.
8. Navigate through the application, performing the actions you want to record.
9. Click the **Stop** button on the Recording toolbar.

The Repository Changes dialog box opens. New or changed windows and controls are displayed.
10. Click **Save** to save the repository changes.

The script is processed and the actions are added to it.

Recording low-level actions

If an application uses non-standard controls, QA Wizard Pro may not recognize those objects and will use low-level action statements to perform keyboard and mouse actions on them. For example, if you are recording a script to test a web page with hover-over menus, use low-level mode to force the recording system to track mouse movements and capture the menu items.

1. While recording a script, click **Low-Level** on the Recording toolbar. Low-level mode is activated.
2. Mouse over an object to use as a starting point.

The selected object is outlined with a red box.
3. Click the selected object.

QA Wizard Pro begins recording low-level actions.
4. Interact with the object.
5. Click **Low-Level** to stop recording low-level commands.

Low-level actions are grouped in an expandable outline. Click the plus sign to view the steps.

Adding new scripts

You can add a new script to the workspace without recording. You may want to add a script to test a function before adding it to an existing script or add an entire script of functions that you call from other scripts.

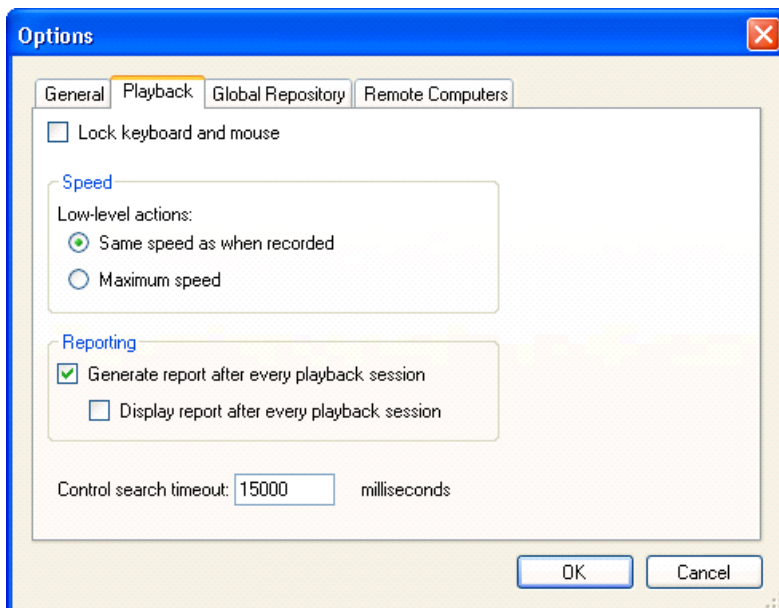
Note: Record a script if you want to add a new script that performs actions in an application.

1. Choose **File > New > New Script**.
The Add New Script dialog box opens.
2. Enter a script **File name**
3. Click **Save**.
The new script is added to the workspace.

Setting playback options

These options control default QA Wizard Pro playback behavior.

1. Choose **Tools > Options**.
The Options dialog box opens.
2. Click the **Playback** tab.



3. Select **Lock keyboard and mouse** to lock the keyboard and mouse when scripts run. You may want to set this option if you are going to be away from the desk while your scripts run or if you are using a remote agent to run scripts.

Note: To unlock the keyboard and mouse while a script is running, press Ctrl+Alt+Delete to open the Windows Security dialog box. Press the Esc to close the dialog box. This returns you to QA Wizard Pro. You can allow the script to continue running or press the Pause/Break key to stop the script.

4. Select a low-level action **Speed** option.
 - **Same speed as when recorded** plays back low-level actions at the record speed.
 - **Maximum speed** plays back low-level actions as quickly as possible. You may want to select this option if scripts are running as expected and you want to speed up playback.
5. Select any **Reporting** options.

- **Generate report after every playback session** generates a script results report after scripts run.
 - **Display report after every playback session** opens the script results report after scripts run.
6. Enter the **Control search timeout** to indicate how long before playback times out.
If the timeout is reached, playback stops and an error displays.

Note: The default value is 15,000 milliseconds.

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

Configuring remote computers

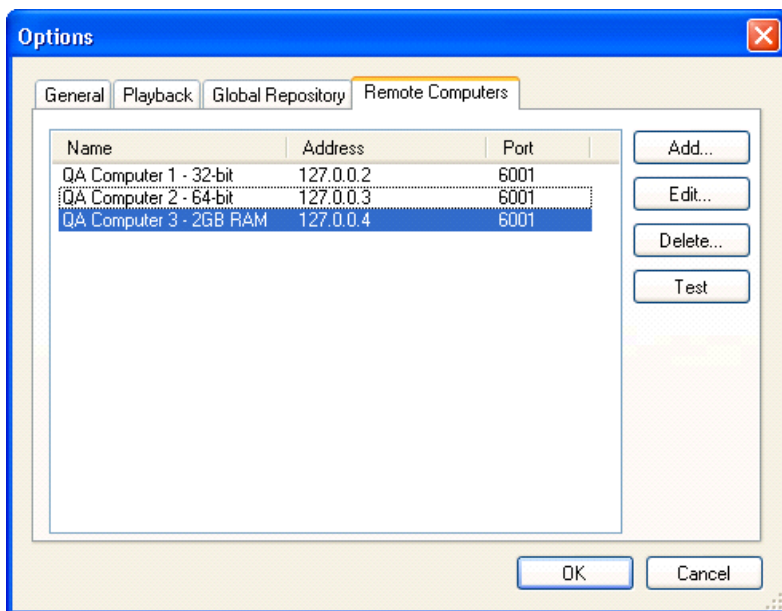
Before you run scripts remotely, you must configure the remote computer options. Keep the following in mind:

- QA Wizard Pro must be installed on the remote computer.
- QAWRemote.exe must be running on the remote computer.
- Each remote computer must have a valid QA Wizard Pro license. It is recommended you use run-time licenses for remote computers.
- The computer you are running remote scripts from must have a full, dedicated QA Wizard Pro license.

1. Choose **Tools > Options**.

The Options dialog box opens.

2. Click the **Remote Computers** tab.



3. Click **Add** to add a remote computer.

- Enter a **Name**. This can be the name of the computer or a name that describes the computer (for example, QA Computer AMD x64).
- Enter the **Hostname** or **IP Address**.

- Enter the **Port Number**. The default is 6001.
 - Click **OK**.
4. Select a computer and click **Test** to test the connection.
If the connections fails, make sure you entered the correct information and retest the connection.
 5. Select a computer and click **Edit** to change the name, hostname, IP address, or port number.
 6. Select a computer and click **Delete** to delete it.
 7. Click **OK** to save the changes.

Note: Scripts use the playback options set on the remote computer, which you may need to change.

Starting QAWRemote.exe

To run your scripts remotely, QAWRemote.exe must be started on the remote computer.

1. Browse to the QA Wizard Pro directory on the remote computer and double-click **qawremote.exe**.

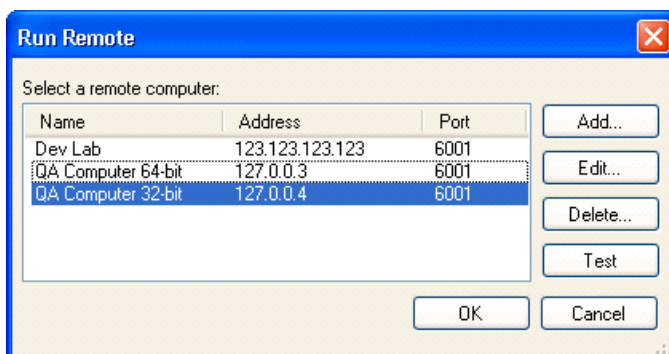
Running scripts remotely

Keep the following in mind before you run scripts remotely:

- If scripts include CallScript statements, the called script name must be a text string.
- If an OpenRecordset function is used to access a local datasheet, the datasheet name must be a text string.
- Variables and other expressions are not supported.
- QAWRemote.exe must be started.

1. Open the script you want to run on the remote computer.
2. Choose **Script > Run Remote**.

The Run Remote dialog box opens.



3. Select the remote computer.
Click **Add** if you need to add a remote computer.
4. Click **OK**.
The script runs on the remote computer. A dialog box opens to confirm the script is running.

Note: If repository variables are defined for the application version, you may be promoted to enter variable values before playback begins.

5. Click **OK** to close the dialog box.
6. When the script finishes running, the run report is saved on the remote computer in C:\Documents and Settings\\My Documents\QA Wizard Reports.

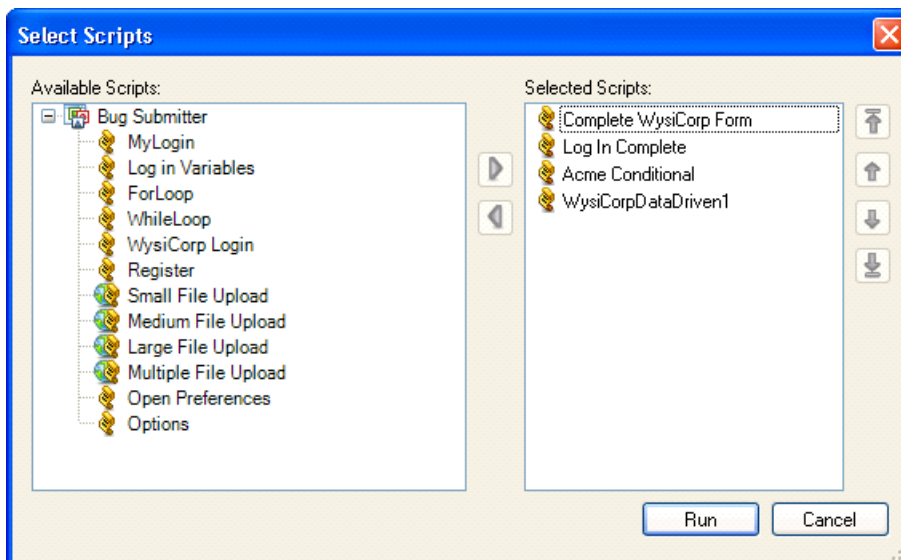
Tip: You can use the QA Wizard Pro status tool to monitor remote scripts. To start the tool, choose **Start > Programs > Seapine Software > QA Wizard Pro > QA Wizard Pro Status Tool** or double click the QA Wizard Pro Status Tool icon on your desktop.

Running multiple scripts remotely

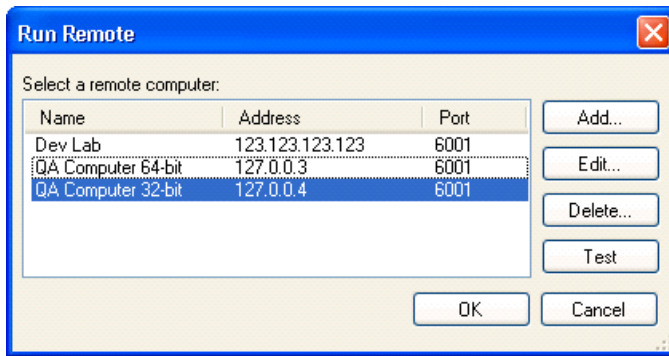
You can run multiple scripts on remote computers. For example, you can run scripts on multiple computers simultaneously to perform tests while you work on other tasks.

Note: QA Wizard Pro must be installed and QAWRemote.exe must be running on the remote computer. To start QAWRemote, double-click the executable in the QA Wizard Pro directory. The remote computer must also have access to the tested application in the global or local application repository.

1. Choose **Script > Run Multiple Remote**.
The Select Scripts dialog box opens.
2. Select the scripts you want to run and click the right arrow button to add them to the Selected Scripts list.



3. Select a script and click **Top**, **Move Up**, **Move Down**, or **Bottom** to reorder the list.
Scripts run in the order displayed.
4. Click **Run**.
The Run Remote dialog box opens.



5. Select the remote computer.

Click **Add** if you need to add a remote computer.

6. Click **OK**.

The script runs on the remote computer. A dialog box opens to confirm the script is running.

Note: If repository variables are defined for the application version, you may be promoted to enter variable values before playback begins.

7. Click **OK** to close the dialog box.

8. When the script finishes running, the run report is saved on the remote computer in C:\Documents and Settings\\My Documents\QA Wizard Reports.

Tip: You can use the QA Wizard Pro status tool to monitor remote scripts. To start the tool, choose **Start > Programs > Seapine Software > QA Wizard Pro > QA Wizard Pro Status Tool** or double click the QA Wizard Pro Status Tool icon on your desktop.

Remote Testing Reports

QA Wizard Pro generates and saves a result report after your scripts finish running on the remote computers. The report includes information about the script such as the test computer configuration, the script name and location, a summary of the test results, and list of warnings if any were encountered. Reports are named using the script name and the date and time the script ran.

Viewing, exporting, and printing reports

1. Choose **View > Reports**.

2. Double-click a report in the Reports pane.

The report opens. It includes information about the script and details about any failures.

3. Choose **File > Print** to print the report.

4. Choose **File > Save As** to save the report in a different format for use with other programs.

Remote Testing Conclusions

The information you gather by running automated testing scripts remotely can be used to improve the reliability of your application or enhance certain features. You may have found that there are many more bugs on the 64-bit version of the application, that it crashes more often on a system with less than two gigabytes of RAM, or that it has some incompatibilities with certain web browsers. By running your testing scripts on computers with various hardware and software configurations, you can better prepare your application for use across a diverse user base.

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) - <http://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/) - <http://www.seapine.com/kb/categories/QA+Wizard+Pro/>