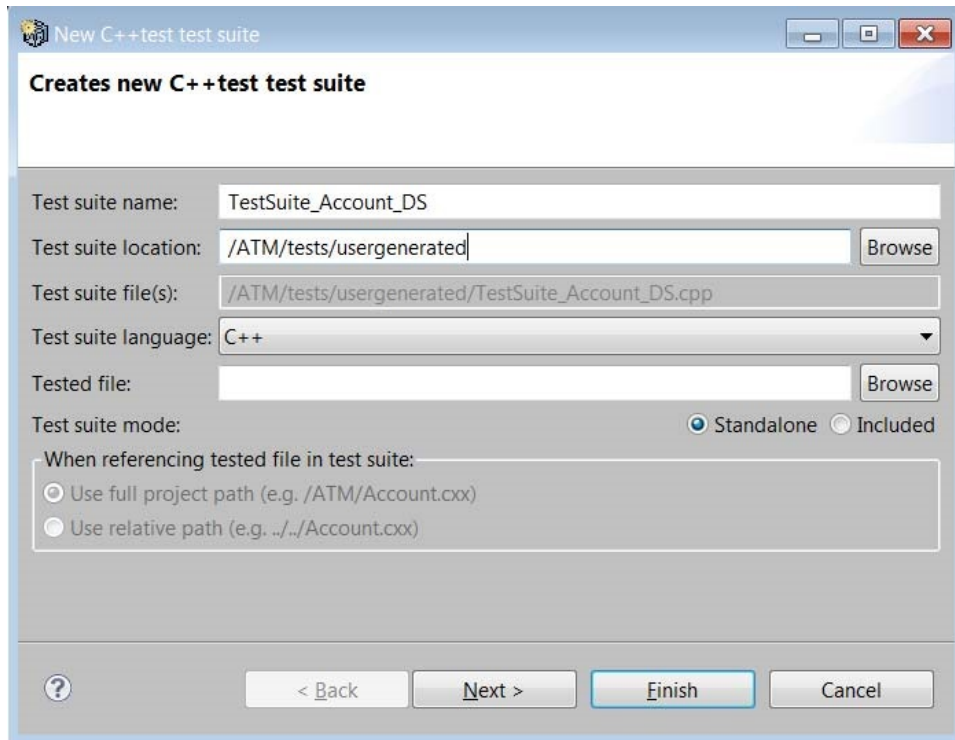


Exercise 12 - Using Data Sources in Unit Tests

1. In the Test Case Explorer, right-click the **tests** directory and choose **Add New> Test Suite**.
2. Set the **Test suite name** and add a usergenerated directory in the **Test suite location**:



3. Click **Finish**.
4. Add a data source to the test suite:
 - a. In the Test Case Explorer, right-click the **ATM** project, then choose **Add New> Data Source**.
 - b. Select **Table** and click **Finish**.
 - c. Enter a **Name**.
 - d. Enable the **First row specifies column names** option.
 - e. Double-click the fields and enter the following values
 - A header: password
 - A2: a1
 - A3: really_long_password
 - A4: foo
 - B header: result
 - B2: a1
 - B3: really_long_password
 - B4: goo.

C++test Report [03/0] TestSuite_Account_DS *New Datasource

General

Name: TestPasswordsDS Type: Table

Rows

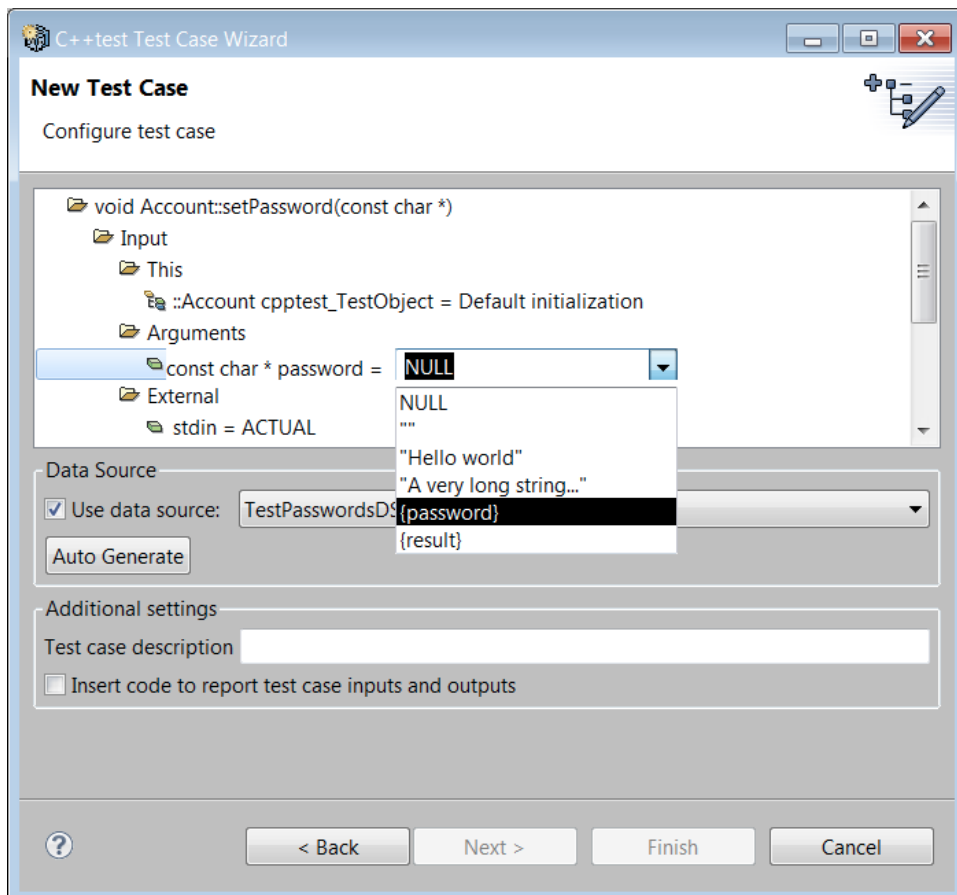
All Range From: 1 To: 1

Table

First row specifies column names

	A	B	C	D
	password	result		
1				
2	a1	a2		
3	really long password	really long password		
4	foo	qoo		
5				

- f. Save the data source file.
5. In the Test Case Explorer, right-click **TestSuite_Account_DS** and choose **Add New> Test Case using Wizard**.
6. Specify the test case:
- In the first page of the wizard, specify the **File** value by clicking **Browse** and navigating to **Account.cxx**. For **Function**, select **void Account::setPassword(const char *)** from the pull down menu.
 - Click **Next**.
 - Check **Use data source** and use the pull down menu to select the **TestPasswordDS** data source created earlier.
 - For **Input> Arguments> password**, double-click on **NULL** and use the pull down menu to select **{password}**.



- e. Click **Finish**.

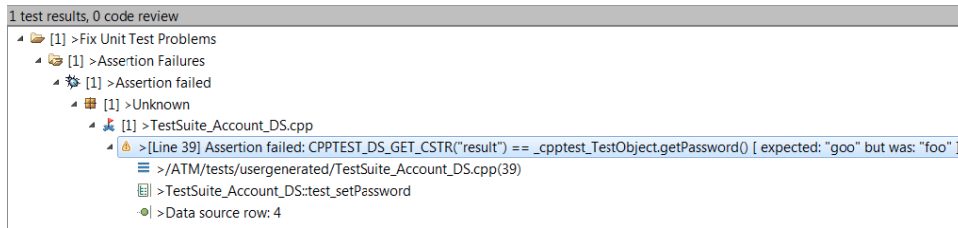
7. Edit test_setPassword test case:

- a. In the Test Case Explorer, double-click the **test_setPassword** test case to open the test suite.
- b. Add **#include "Account.hxx"** to the top of the test suite.
- c. Remove the three post-condition checks at the end of the test_setPassword test case and replace with `CPPTTEST_ASSERT_CSTR_EQUAL(CPPTTEST_DS_GET_CSTR("result"), _cpptest_TestObject.getPassword());`

```
/* CPPTTEST_TEST_CASE_BEGIN test_setPassword */
/* CPPTTEST_TEST_CASE_CONTEXT void Account::setPassword(const char *) */
void TestSuite_Account_DS::test_setPassword()
{
    /* Pre-condition initialization */
    /* Initializing argument 0 (this) */
    ::Account_cpptest_TestObject ;
    /* Initializing argument 1 (password) */
    const char * _password = CPPTTEST_DS_GET_CSTR("password");
    /* Tested function call */
    _cpptest_TestObject.setPassword(_password);
    /* Post-condition check */
    CPPTTEST_ASSERT_CSTR_EQUAL(CPPTTEST_DS_GET_CSTR("result"), _cpptest_TestObject.getPassword());
}
/* CPPTTEST_TEST_CASE_END test_setPassword */
```

- d. Save the changes to the test suite.

- 8. Run the tests by selecting **test_setPassword** in the Test Case Explorer, then choosing **Parasoft> Test Using> User-Defined> Run Unit Tests (Project Scope)**. The Test Progress Tab will report 3 tests passed and 1 assertion failures.
- 9. Open the Quality Tasks view and expand **Fix Unit Test Problems** to review the assertion failure that we built into the Data Source table.



10. Fix the result in the table and rerun the test to observe the correction.

You can add additional tests by simply adding or modifying rows in the data source.