Updating Tests as Services Change 1

Parasoft SOAtest has two capabilities that help you make bulk updates to your tests when the services that they are testing change:

- **Change Advisor** identifies changes to a service that impact any tests created against a different version of the service and allows you to update your tests in bulk without having to manually modify them one by one.
- **Search and Replace** allows you to search your tests for values that you know have changed in your service and update them all at once with a new value.

These two capabilities are typically used in sync to ensure rapid, accurate updating. This lesson will show you how to use both capabilities to update tests.

## Creating Tests Against an "Old" Version of a Service

Start by creating a set of tests for an older version of a bookstore service. We'll later update these tests using the Change Advisor and Search and Replace capabilities.

To create tests vs. an old version of the bookstore service:

1. Create a new empty project and name it `Change Advisor Lesson`:
    a. Choose **File> New> Project**.
    b. Select **General> Project**.
    c. Click **Next**.
    d. Enter `Change Advisor Lesson` under **Project Name**.
    e. Click **Finish**.
2. Add a new .tst to that project:
    a. Right-click the **Change Advisor Lesson** node in that added project and choose **Add New> Test (.tst) File**.
    b. Enter `Change Advisor Lesson` under **File name**, then click **Next**.
    c. Choose **SOA> WSDL**, then click **Next**.
    d. Type `http://localhost:8080/parabank/services/store-01?wsdl` as the **WSDL URL**.
    e. Ensure that **Create Functional Tests from the WSDL** is checked.
    f. Click the **Finish** button.
3. Double click **Test Suite: Test Suite> Test 2: addNewItemToInventory** to open its editor.
4. Check the **book** element and click the **Edit** button next to it.
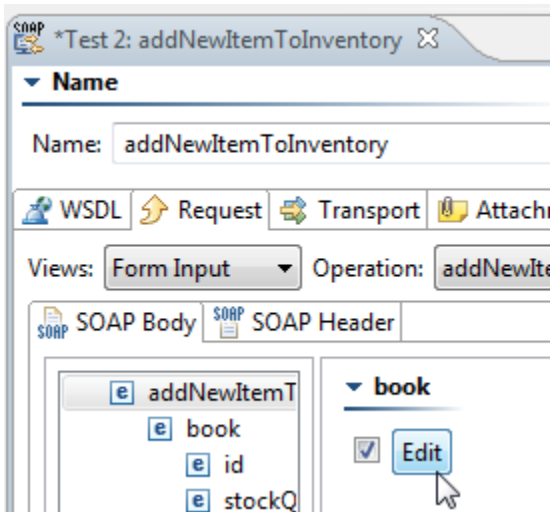

5. Fill in the following fields:
    - **Id**: Enter `10`
    - **Price**: Check the **price** box, then enter `9.99`
   - **stockQuantity**: Enter `5`
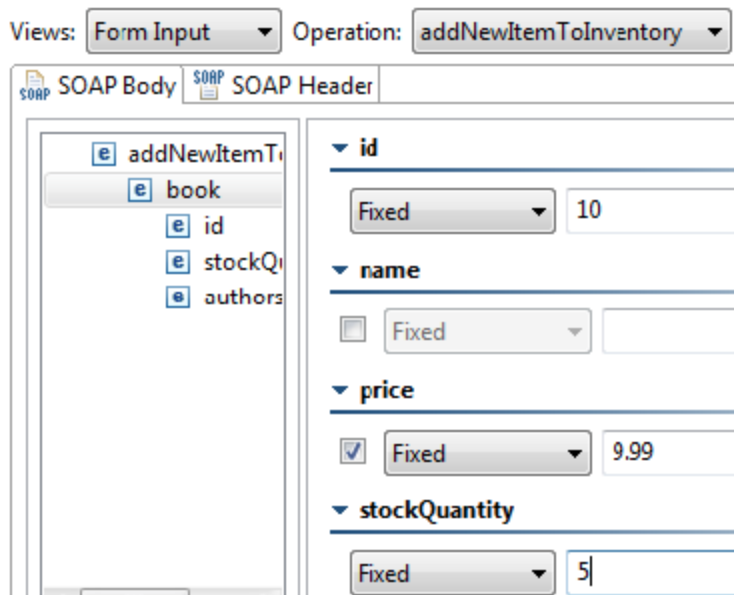
6. Click the **Save** toolbar button. Note that when we update our tests to test the newer version of the bookstore, we will confirm that these values are carried over properly.
7. Select **Test 2: addNewItemToInventory** in the Test Case Explorer and run it.
8. Click the **Add test or output** toolbar button.
9. Select **Response> SOAP Envelope> XML Assertor**, then click **Finish**.
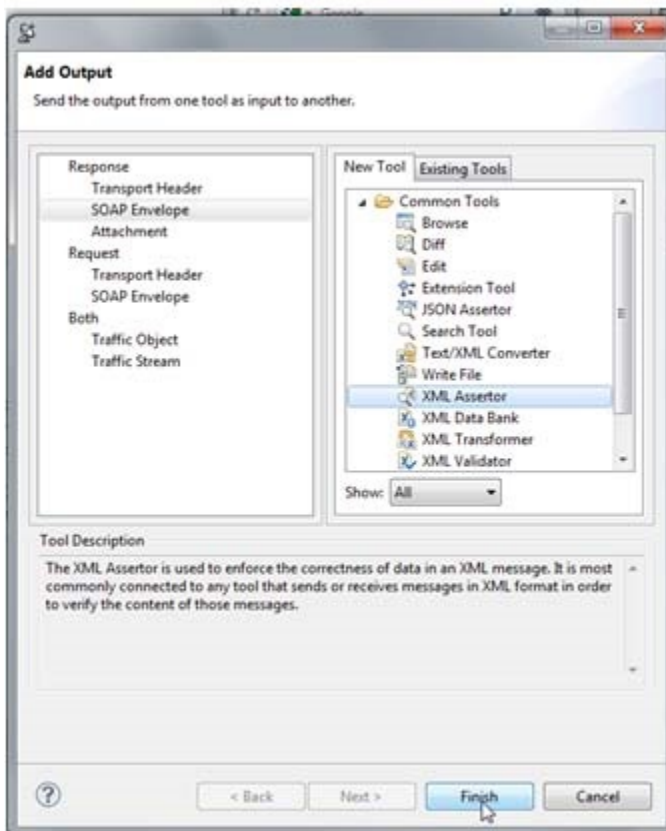
10. In the Assertor's editor, open the **Configuration** tab, click **Add**, then select **Structure Assertions> Occurrence Assertion**, and click **Next**.
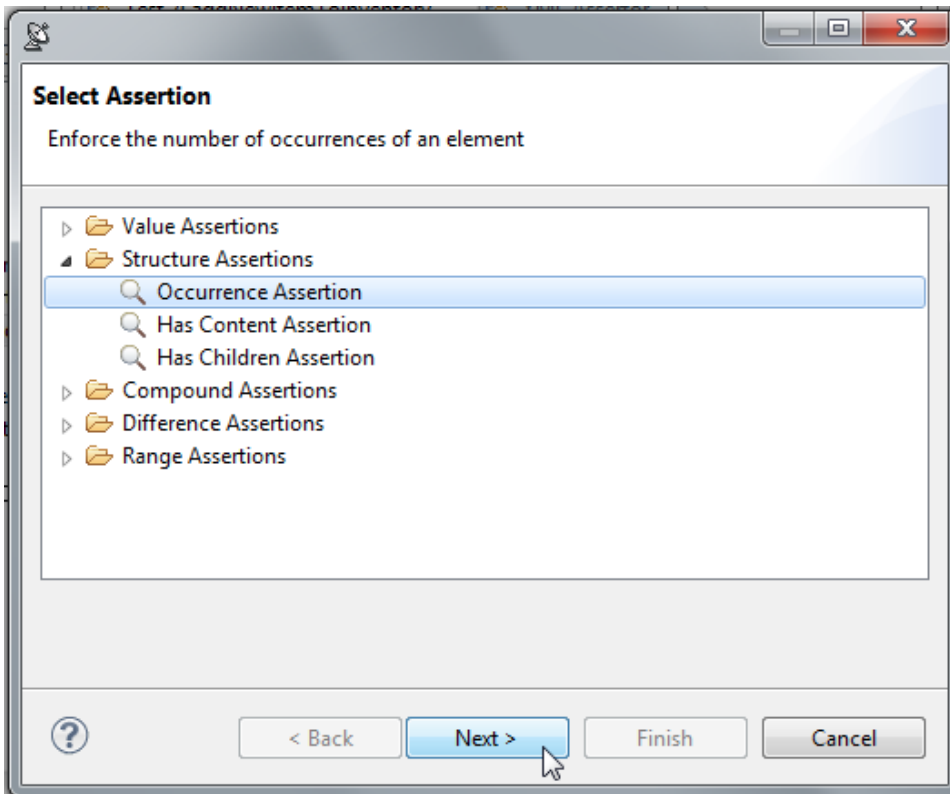
11. In the XML tree shown, select **price**, then click **Finish**.
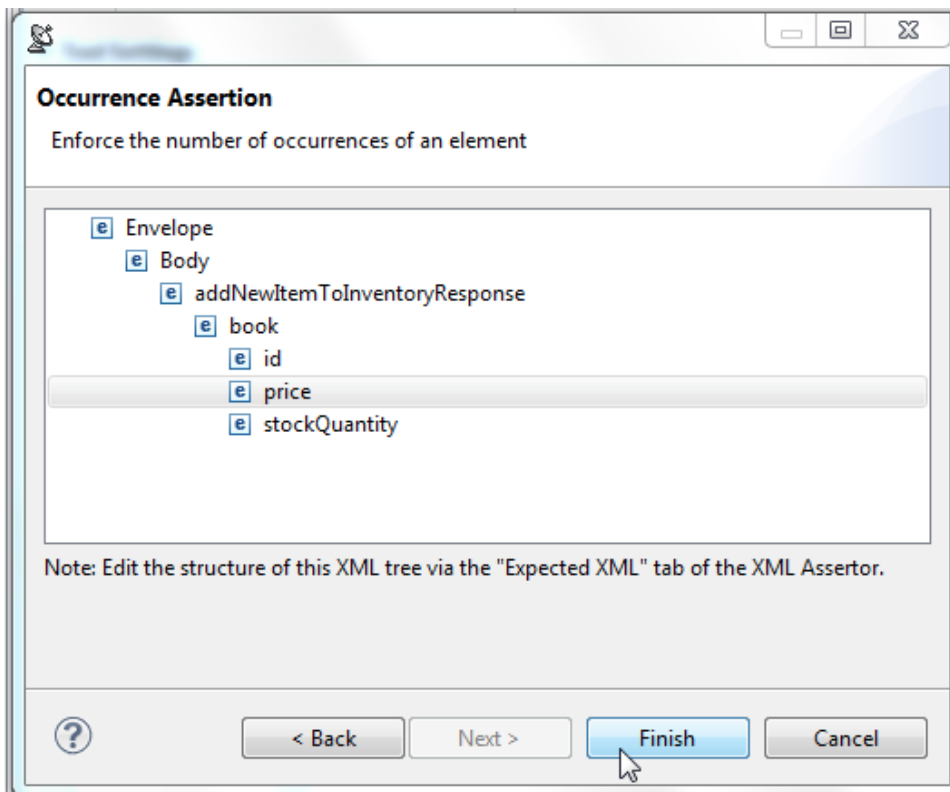
12. Under Occurrence Assertion Configuration, make sure that **==** and **Fixed** are selected in the drop down menus, then enter 1 in the input field.

▼ Occurrence Assertion Configuration

Occurrences of element must be ▯ == ▾ ▯ expected value: ▯ Fixed ▾ ▯ ▯ 1 ▯

Using this assertor, we are declaring that the element `price` occurs exactly once in the SOAP response message. If this element is later changed in the updated version of the service, we will have to re-configure the updated test's XML Assertor.
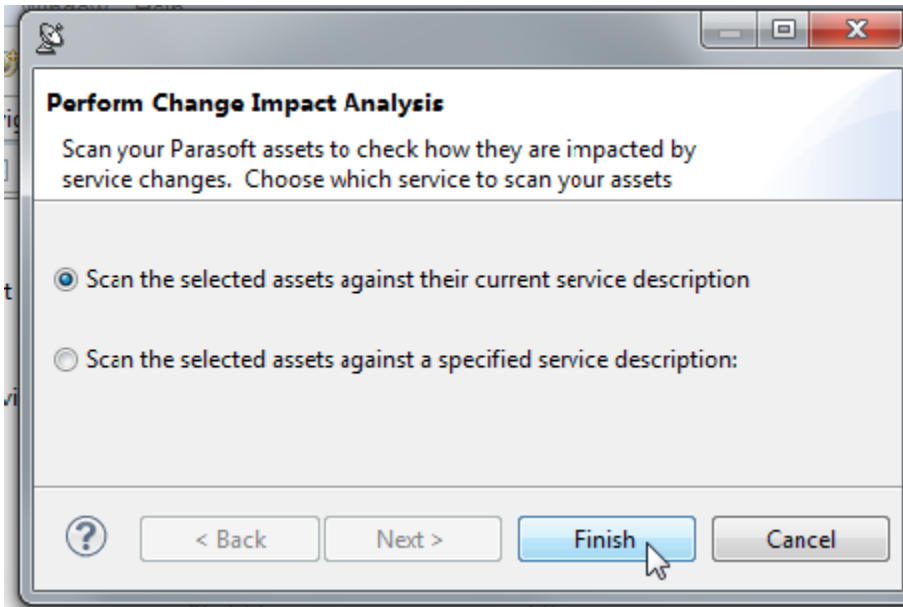
Performing Change Impact Analysis

To perform change impact analysis on these test assets:

1. Open the Change Advisor perspective by selecting **Window> Perspective> Open Perspec-tive> Parasoft Change Advisor**.
2. In the Test Case Explorer, select **Change Advisor Lesson> Change Advisor Lesson .tst> Test Suite: Test Suite**.
3. Click the **Perform Change Impact Analysis** toolbar button.
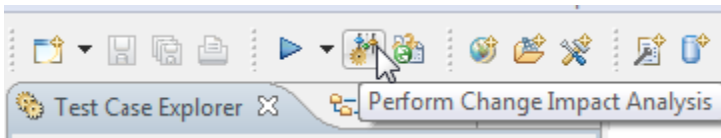
4. In the dialog that opens, select **Scan the selected assets against their current service description**, then click **Finish**.

5. In the Quality Tasks view, change the layout to the SOAtest Change Impact Layout, which is designed for reviewing the results of running the change impact analysis. You can switch layouts by clicking the Quality Tasks view's **SOAtest Change Impact Layout** toolbar button...
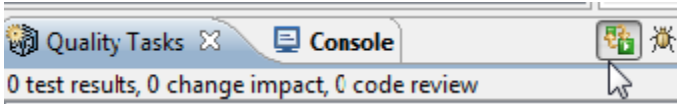
or by selecting it from the drop-down menu.
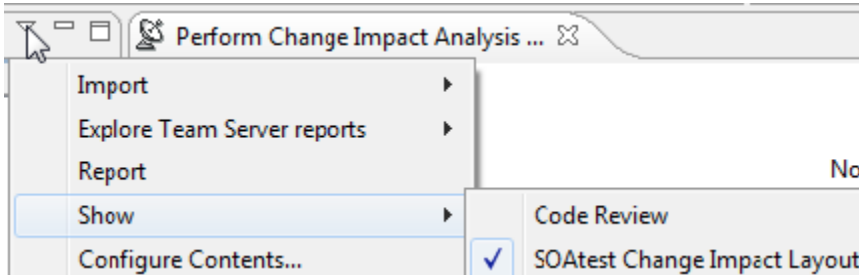
> ⓘ **No Change Impact Layout toolbar button?**
>
> If the green SOAtest Change Impact Layout toolbar button is not present, choose the Configure Contents command, then check the **SO Atest Change Impact Layout** checkbox.

This Change Impact Analysis was configured to compare the service information saved in the tests to the service information contained in the "live" WSDL that the tests currently reference. It would report any WSDL changes that would impact the test cases. Since all of our tests were created with this WSDL just minutes ago (and the WSDL has not changed since then), no changes are reported by the analysis. Next, we'll see what happens if the change impact analysis is run vs. a different WSDL.

6. In the Test Case Explorer, select **Change Advisor Lesson> Change Advisor Lesson.tst> Test Suite: Test Suite**.
7. Click the **Perform Change Impact Analysis** toolbar button.
8. Select **Scan the selected assets against a specified service description**, ensure that **WSDL** is selected under Artifact type, then complete the dialog as follows:
   - **Scan Assets that use the following WSDL**: `http://localhost:8080/parabank/services/store-01?wsdl`
   - **Compare assets to the following WSDL**: `http://localhost:8080/parabank/services/store-01V2?wsdl`
     The first option that tells SOAtest which tests to perform the analysis against. The second option tells SOAtest what WSDL to compare the tests against. The results will now report changes as a result of the service differences found in these WSDLs.
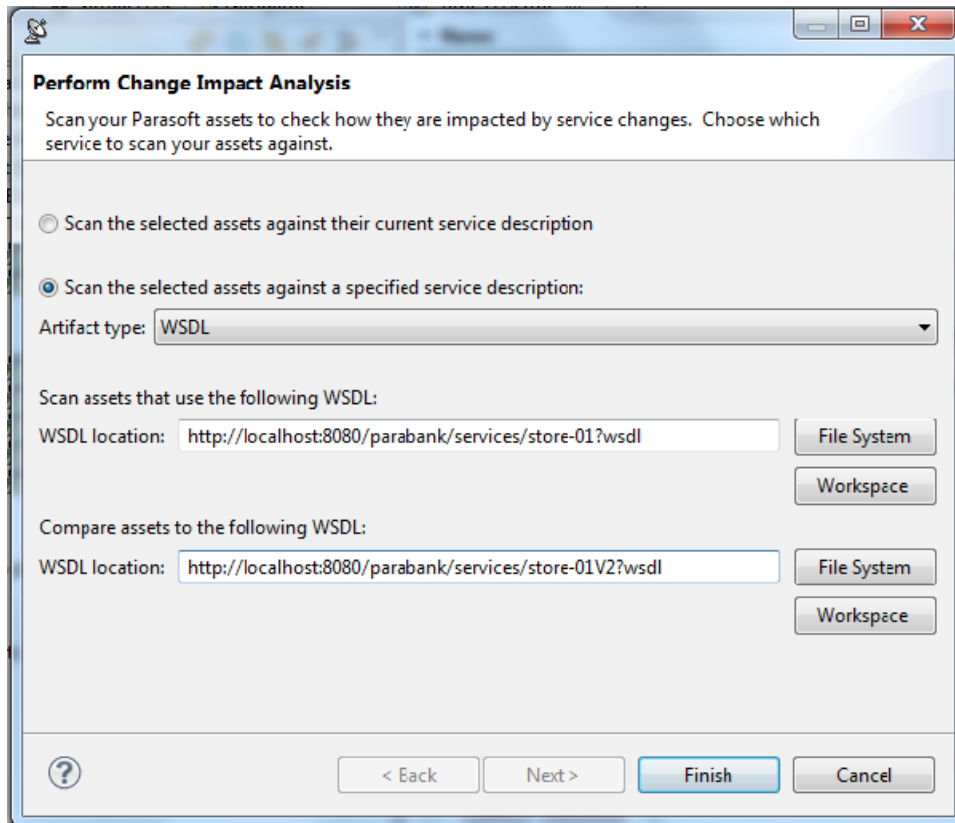
9. Click **Finish**.
10. In the Quality Tasks view, explore the changes that were reported. For many of the tests, the namespace and targetURI changed.
    - Review the results for Test 2: addNewItemtoInventory.
    - Notice that there are a number of other change reported:
        - The new element `productInfo` was added
        - The name of the element `price` changed
   - The order of elements in the message changed
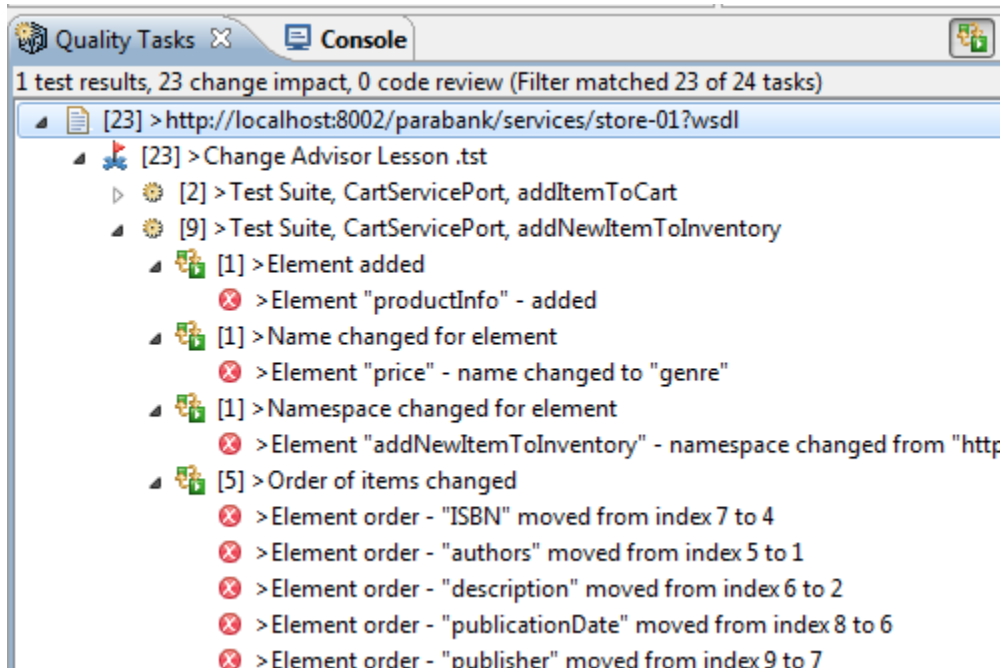
Note that the Change Impact results do not show all changes that are in the service definition; they show only service definition changes that actually affect the tests. For example, if there were changes in operations that were not currently being tested, then no change impact results would be reported for those changes.
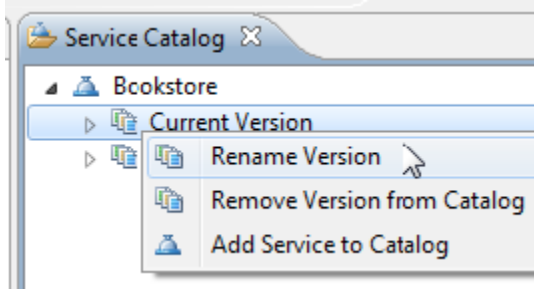
Creating a Change Template

When the Change Impact analysis is run, the Service Catalog view is automatically populated with information about the services and versions of the services that are being tested by the tests being analyzed.

Notice how the store service and versions have already been populated. By default SOAtest does not know the official names of the services and versions being tested, so it uses the service definition as the service name. It also uses the name "Old Version" to represent the version of the service that the tests are currently testing, and it uses the name "Current Version" to represent the version represented by the new service definition. The Service Catalog allows you to browse the operations that were detected for each version.

1. Rename the service and versions to something that we can better understand:
    a. In the Service Catalog, right-click the top-level service node, then choose **Rename Service**.
    b. Enter `Bookstore` as the service name.
    c. Right-click the **Old Version** node for the bookstore service, choose **Rename Version**, then replace `Old Version` with `1.0`.
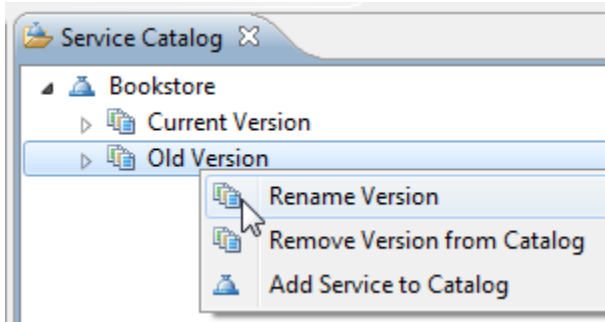

    d. Right-click the **Current Version** node for the bookstore service, choose **Rename Version**, then replace `Current Version` with `2.0`.
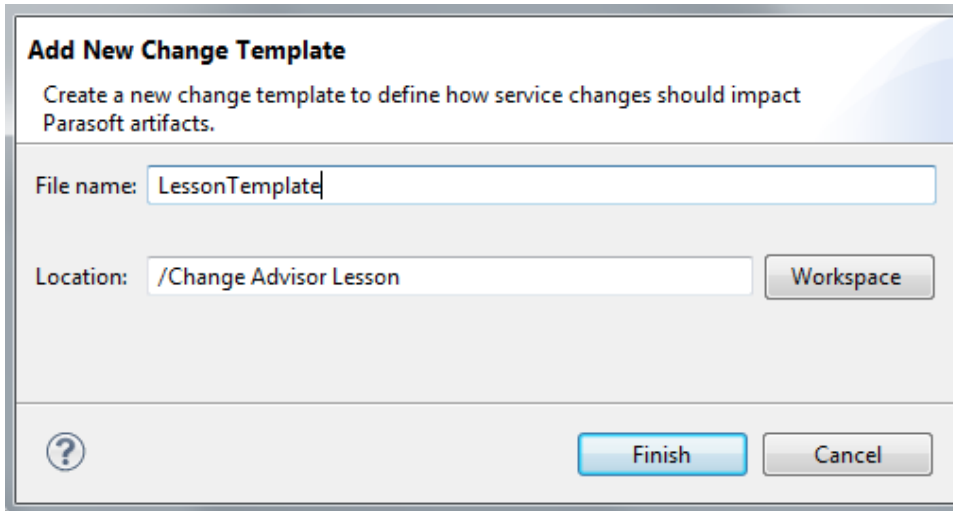

    Next we want to define a change template that will define how we should update our tests so that they test the new version of the service.
2. Create a change template as follows:
    a. In the Service Catalog, select the nodes **1.0** and **2.0**.
   b. Right-click the selection, then choose **Add Change Template 1.0 -> 2.0**.
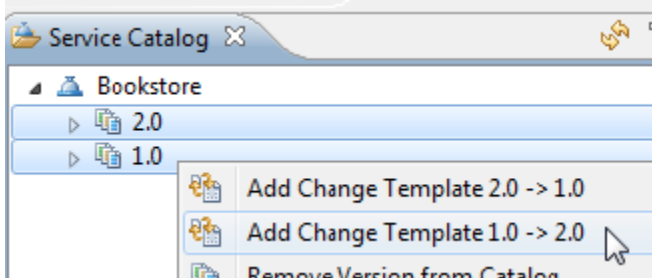
    c. Enter `LessonTemplate` as the file name, then leave `/Change Advisor Lesson` as the location.

Add New Change Template

Create a new change template to define how service changes should impact Parasoft artifacts.

File name: LessonTemplate

Location: /Change Advisor Lesson   [Workspace]

[Finish] [Cancel]

    d. Click **Finish**. The change template editor will open.
3. Review the first tab of the change template that was created. Note that:
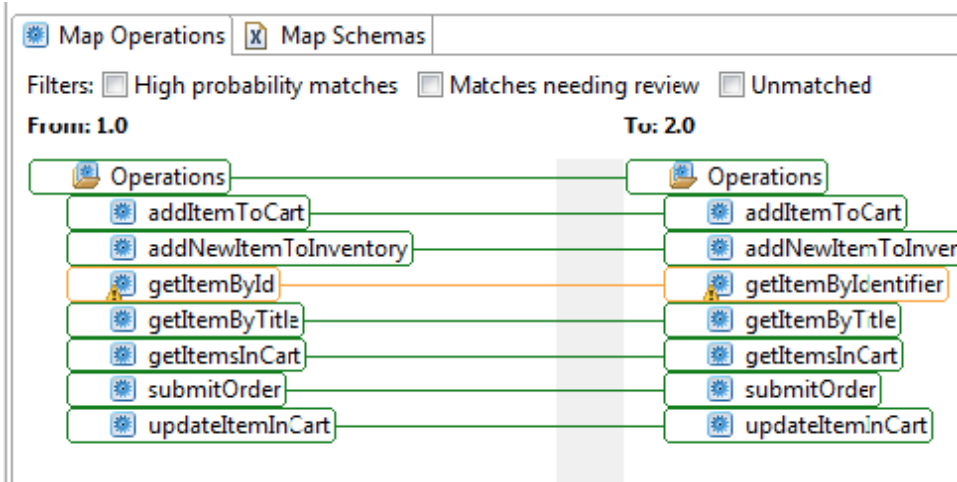
- The Map Operations tab displays the operations defined in both WSDLs, as well as a default mapping between the different versions of the WSDL. When SOAtest first creates a change template, it creates a default mapping between the different versions, attempting to match nodes that it thinks are the same.
- Green matches denote the fact that SOAtest is confident that the mapping it made is correct, and review is not required.
- Yellow matches denote that SOAtest is less confident that the mapping it made is correct, and review is recommended to confirm or update the mapping.
- Here, all operation mappings are shown in green—except for the mapping between the operation `getItemById` and `getItemByIdentifier`. It appears that this operation changed names, but that the meaning of the operation is the same. This seems like a good match.

4. Indicate that we've reviewed the mapping between the operation `getItemById` and `getItemByIdentifier` by right-clicking **getItemById** and choosing **Mark Match getItemById -> get-ItemByIdentifier as Reviewed**". This will make the match turn green.
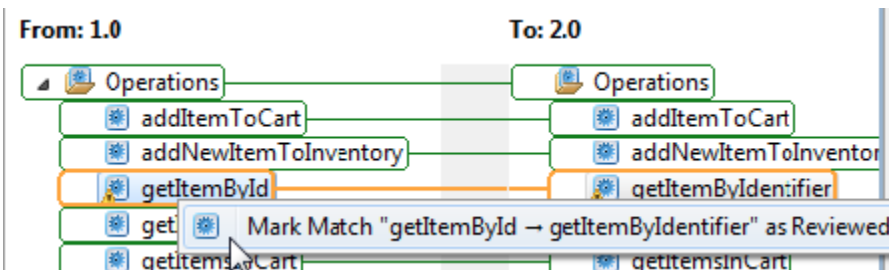

5. Review the second (Map Schemas) tab of the change template that was created. Note that:
   - This tab displays the schemas defined for each operation. You can double-click on the editor tab to cause it to expand so that the editor is larger.
   - The `addNewItemToInventory` operation is marked with a red 'X', indicating that there are differences in the schema for that operation that need to be reviewed.
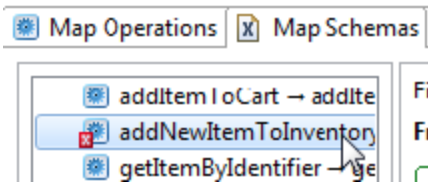
- Other operations do not need review since SOAtest did not detect any changes in those operations in need of review. If you select each of the operations without the red 'X' icon, you'll see that all matches are in green.
6. Select the **addNewItemToInventory** operation to explore it. Note that
    - SOAtest matches the `price` and `genre` elements, but indicates that they need review
   - The operation contains a new element named `amount` that did not appear in the original version.
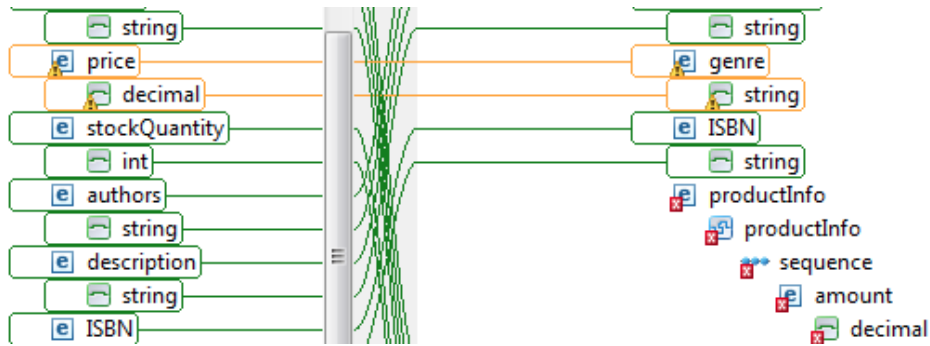
7. Indicate that the mapping between the `price` and `genre` elements is incorrect by selecting the **price** element and clicking **Disconnect**.
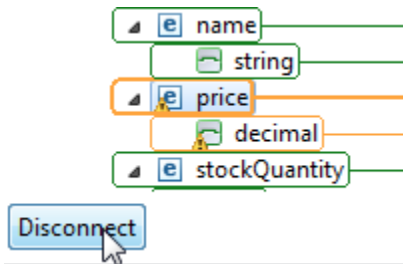

It appears that the `price` element was renamed to `amount`, and that `genre` is the new element in the WSDL.
8. Indicate that the `price` element was renamed to `amount` by select **price** and **amount**, then clicking the **Connect** button.
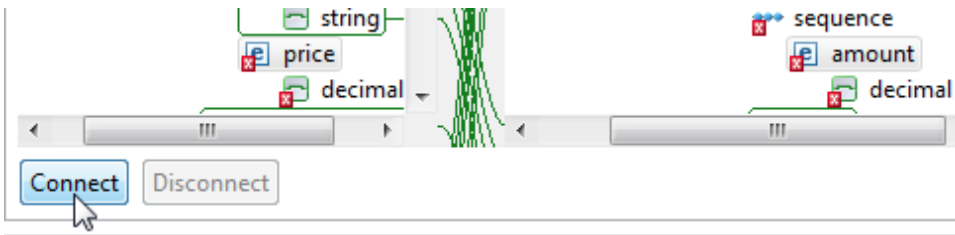

The `genre` element is a new element that appears for this operation. In order for our tests to be updated properly, we must configure a default value for that element that will be configured in each of the updated tests. Otherwise the updated tests would send an empty value for that new element.
9. Configure a default value for the `genre` element as follows:
    a. Right-clicking **genre** and choose **Configure Default Value for Added "genre" element**.
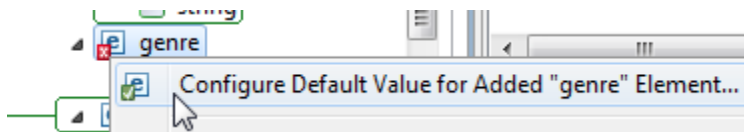
   b. In the dialog that opens, enter `Literature` in the input field, then click **OK**.
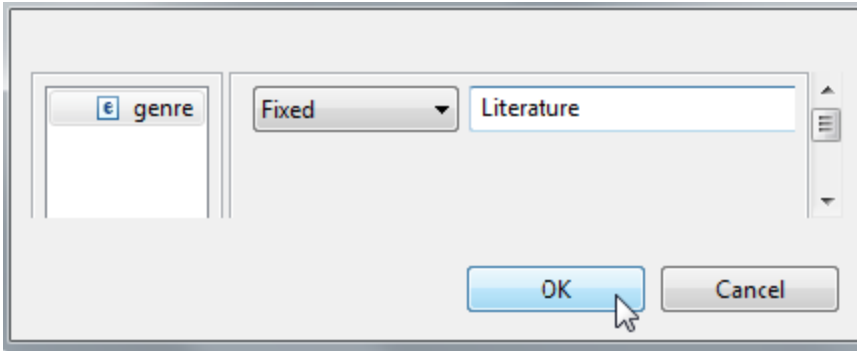
    c. Notice that `genre` now has a green '?' and that all matches and unmatched nodes are now green. This indicates that we're finished reviewing and configuring this operation.

10. Indicate that the three parent nodes of amount have been reviewed by right-clicking the highest **productInfo** element and choosing **Mark"produc tInfo" and Children as Reviewed**.

11. Save the template by clicking the **Save** toolbar button. At this point, our template has been defined and is ready to be applied to our tests. Double-click the change template editor tab to bring it back to normal size.

Applying the Change Template to Update Tests in Bulk

To update the tests impacted by this service change:

1. In the Test Case Explorer, select **Change Advisor Lesson.tst**, then click the **Refactor Using a Change Template** toolbar button.
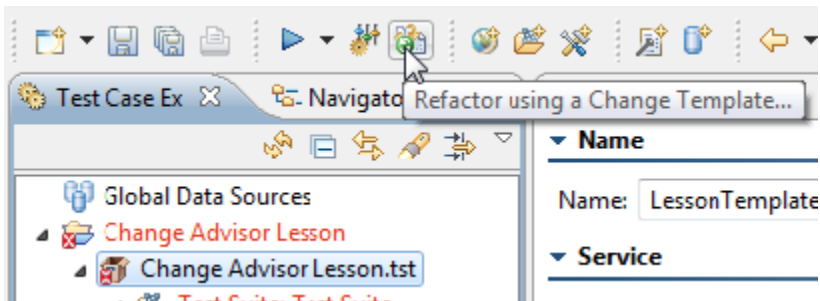
2. Complete the wizard as follows:
    a. Complete the first page as follows, then click **Next**.

Version: 1.0
Update Parasoft assets to use service version: 2.0
By applying the change template: LessonTemplate

b. In the second page, leave **New Location** selected, then use the **Workspace** button to select the Change Advisor Lesson project. This will cause the updated tests to be saved in a new folder within the Change Advisor Lesson project. Note that, in some cases, you might want to overwrite your previous tests. However, this is highly discouraged unless you have the tests backed up (e.g., using a source control system). Otherwise, you will lose your original tests.

c. Click **Finish**. A new folder called "Change Advisor Lesson" will be created. This folder contains the updated tests.
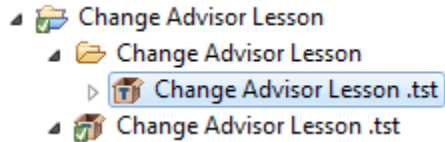

3. Open the .tst file, and open the editor for **Test Suite: Test Suite> Test Suite: CartService-Port> Test 2: addNewItemToInventory**. Note that:

- If you select the **productInfo** element, you'll see that the value that we had configured for the `price` element has been transferred to the amount element, as defined in our change template.
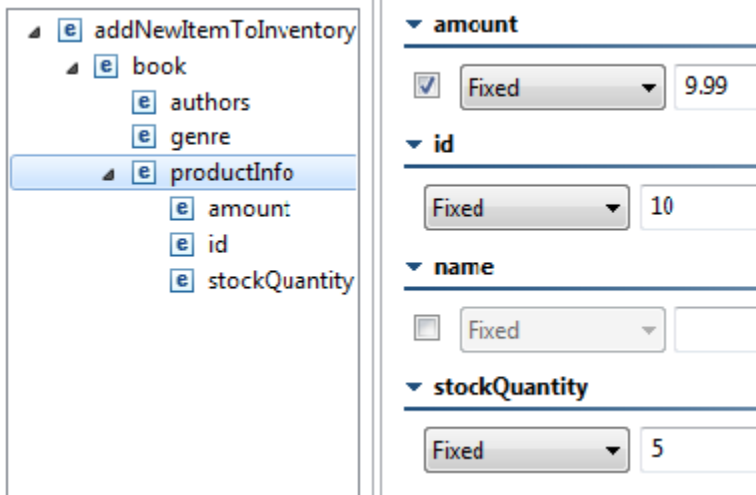
- The other values that we inputted earlier have carried over to the new test. Select the genre element, and note that the value `Literature` appears there, as defined in our change template.
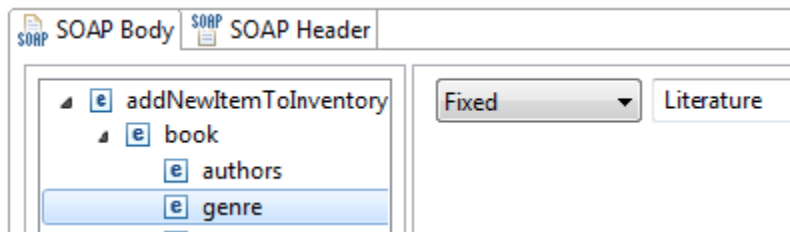
- A new, updated environment has been added to the **Environments** node of the new folder, and it has been set as the active environment.
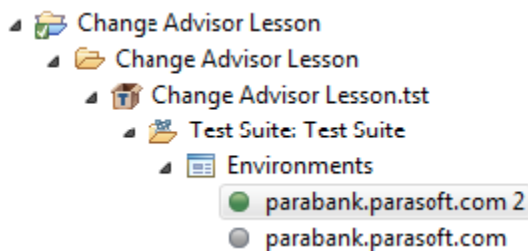
4. Select the new folder **Change Advisor Lesson> Test Suite: Test Suite**, click the **Run Change Impact Analysis** toolbar button, select **Scan selected assets against their current service description** radio button, then select **Finish**. Note that no changes are reported, since the new tests are now updated to test the current version of the service.

Updating Tests Using Search and Replace

Change Advisor is applicable for tests that are configured:

- Against one of the supported artifact types, and
- In Form Input mode.

However, there are cases where it is not applicable. For example, messaging clients that are in Form XML are not supported by Change Advisor. XPaths in XML Assertors also do not get updated by Change Advisor. To support such cases, SOAtest provides a Search and Replace functionality that can be used to update these kinds of fields in bulk. Otherwise, it would be tedious to update fields one-by-one in cases where the same modified values are used over and over.

To use search and replace to update values:

1. Right-click the new version of **Change Advisor Lesson.tst**, and choose **Test Using Example Configuration**.
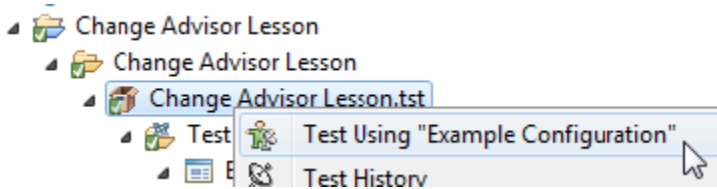

2. In the Quality Tasks view toolbar, click the **SOAtest Default Layout** button to display the functional test results.

3. Look at the error reported for Test 2: addNewItemToInventory, and note that the failure is occurring because the `price` element no longer appears in the response XML (remember that it was renamed to `amount` and now appears under the `productInfo` element).
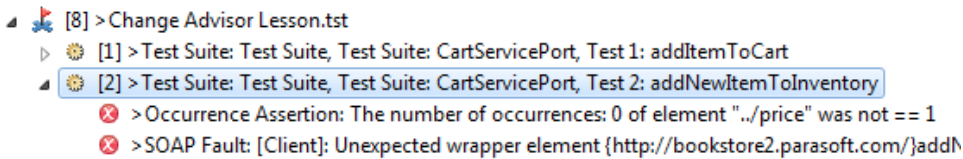
4. Select the new **Change Advisor Lesson.tst** in the new Change Advisor Lesson folder, then choose Search> Parasoft Search from the toolbar.
5. Enter book/price in the **Containing text** field, select **XPath** from the **Field type** selection box, enable **Selected resources**, then click **Search**.
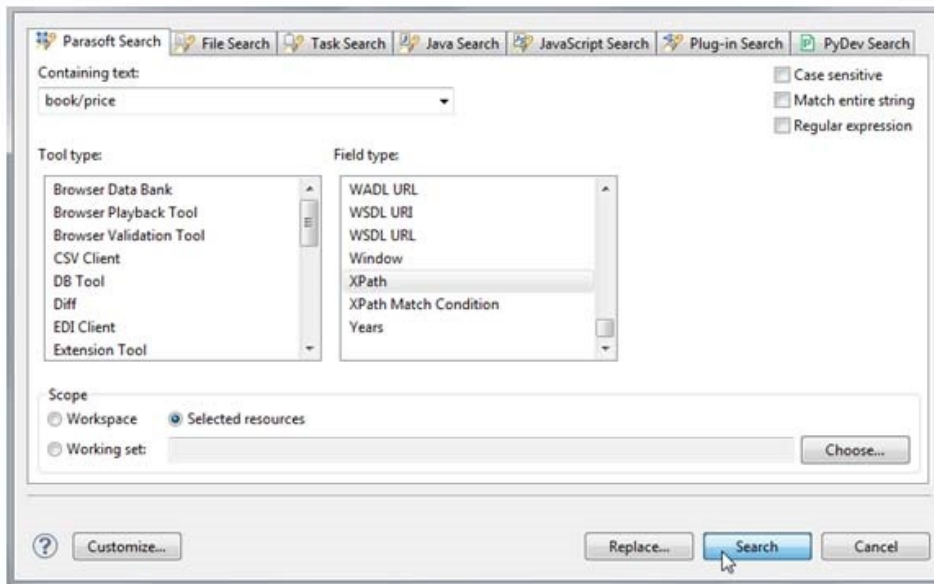
The Search view will populate with the search results.
6. Right click the XPath underneath the XML Assertor, then choose **Replace Selected**.
7. In the dialog that opens, enter `book/productInfo/amount` in the **With** field, then click **OK**.

8. Select **Test 2: addNewItemToInventory** in the folder created for our new WSDL, increment the **id** field, and select the **Run Tests** toolbar button. The test should now succeed since the XPath in the XML assertion is now up to date.