

Working with Fixed Length Messages

SOAtest provides flexible fixed length message format support. Since many different fixed length formats exist (and there is no single standard), Parasoft SOAtest is easily configured to recognize any fixed length message format that the organization works with. This enables you to take advantage of SOAtest's rich interface for message configuration, sending, and validation when working with very specialized message types.

In order to configure and validate fixed length messages in SOAtest, you need to perform the following one-time configuration:

1. Define a data model that describes your fixed length data format.
2. Register the data model so that SOAtest can use it in the Fixed Length Client (and other applicable tools).

At this point, you can use the data model in the Fixed Length Client, Fixed Length Call Back Tool, and XML Converter. You can configure the message in the Fixed Length Client and validations in outputs of the Fixed Length Client.

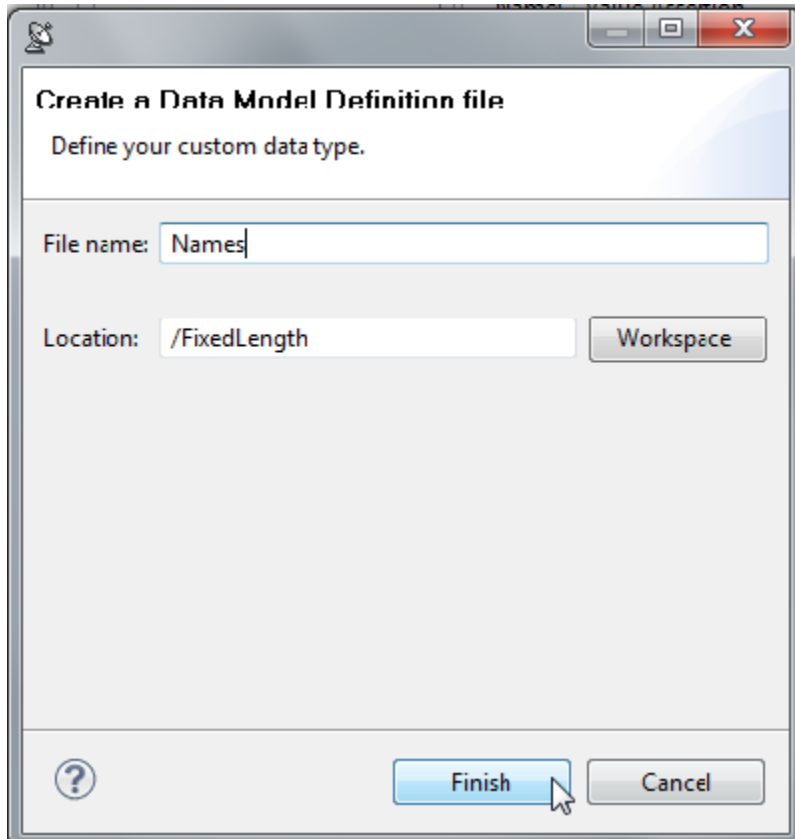
Creating a Fixed Length Data Model

First, we need to understand the message format for which we are going to create a data model. Consider a message where each record is on a different line and simply contains two fields: one for last name and one for first name. Each field is 10 characters long, is right-aligned, and is padded by spaces.

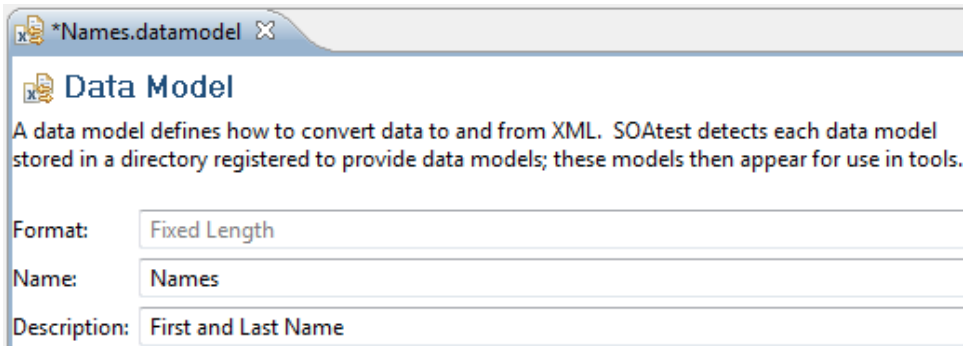
```
Smith   John
Doe     Jane
```

To create a fixed length data model for this format:

1. Create a new empty .tst file as follows:
 - a. Choose **File> New> Project**.
 - b. Select **SOAtest> Empty Project**, then click **Next**.
 - c. Enter a project name (FixedLength), then click **Finish**.
2. Create a new Data Model Definition file as follows:
 - a. Go to **File> New> Data Model Definition File**.
 - b. Enter `Names` as the file name, and choose `/FixedLength` as the location.



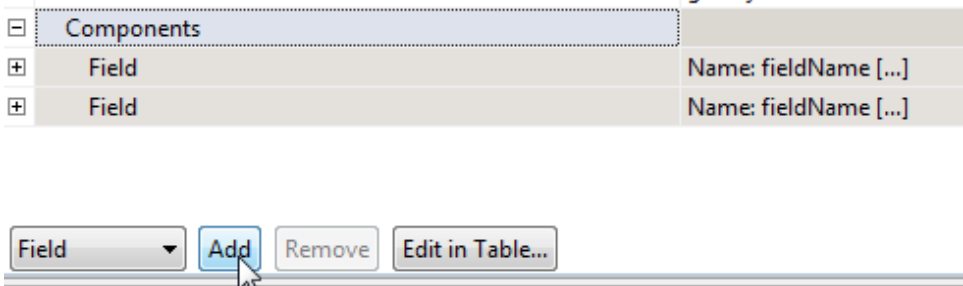
- c. Click **Finish**. The Data Model editor will open.
- d. Enter `Names` in the **Name** field, and `First and Last Name` in the **Description** field.



3. Expand the General section and specify the following:
 - **Name** - people
 - **Record name** - person
 - **Record separator** - {platform newline}

[-] Section	
[-] General	
Name	people
Record name	person
Record separator	{platform newline}
Criteria for next record	greedy

4. Select the **Components** row, then click the **Add** button (at the bottom of the editor) twice to add two fields to the data model.

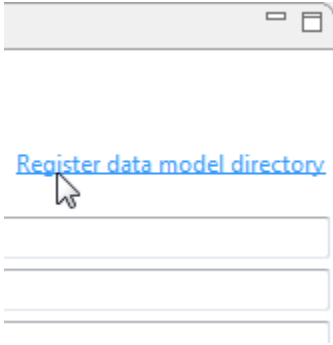


5. Expand the first field and specify the following:
 - **Name** - LastName
 - **Alignment** - right
6. Expand the second field and specify the following:
 - **Name** - FirstName
 - **Alignment** - right

[-] Field	
Name	LastName
Type	string
Length	10
Alignment	right
Padding	{space}
[-] Field	
Name	FirstName
Type	string
Length	10
Alignment	right
Padding	{space}

7. Save the Data Model editor.

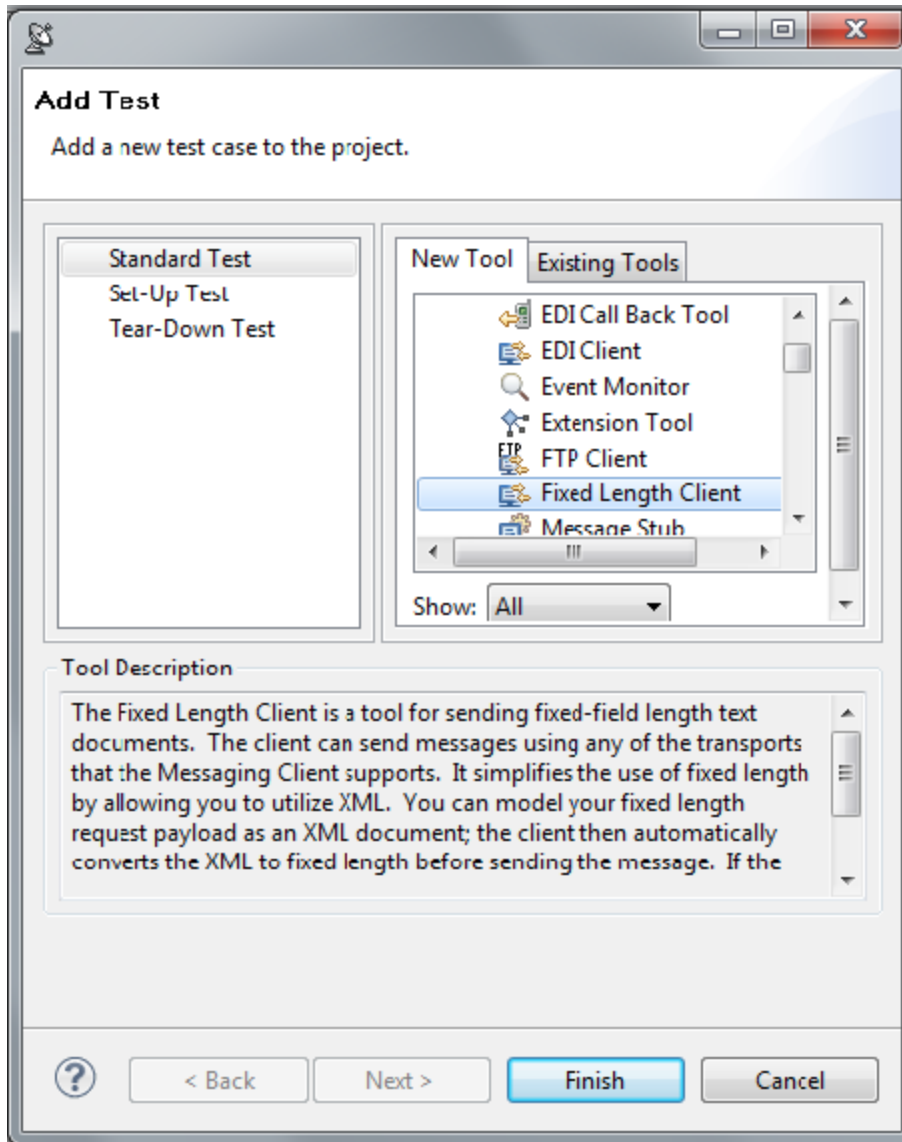
8. Click the **Register data model directory** link in the top right corner of the editor. This registers the data model that we just created with SOAtest. The model is now available to the appropriate tools.



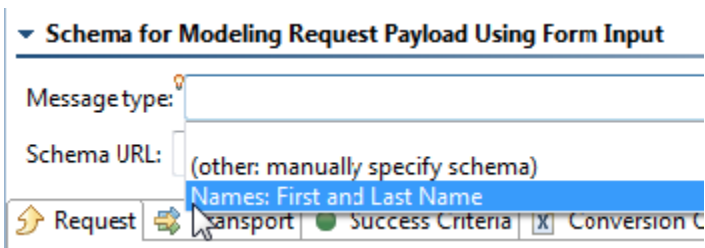
Creating a New Fixed Length Client

To create a new Fixed Length Client

1. Find the FixedLength.tst file that you created in the previous exercise.
2. Create a new Fixed Length Client as follows:
 - a. Right-click the test suite node, then choose **Add New> Test**.
 - b. Select **Fixed Length Client**, then click **Finish**.



3. Select the data model that we just created by choosing **Names: First and Last Name** in the **Message type** combo box.



4. In the **Request** tab, ensure that Input Mode is set to **Form Input**.
5. Modify the automatically-generated **person** element to use **Smith** as the last name and **John** as the first name.

Request Transport Success Criteria Conversion Options

Input Mode: Form Input Element: people {}

people	Fixed	Inhn
person		
LastName		
FirstName		

6. Add a second **person** element by selecting the **people** element, then clicking the **Insert** button below the table.

Request Transport Success Criteria Conversion Options

Input Mode: Form Input Element: people {}

people	▼ person	
person	Index	
LastName	0	
FirstName		
	Insert	Delete Modify

Select the new **person** element, then enter *Doe* as the last name and *Jane* as the first name.

Request Transport Success Criteria Conversion Options

Input Mode: Form Input Element: people {}

people	▼ LastName	
person	Fixed Doe	
LastName		
FirstName		
person	▼ FirstName	
LastName	Fixed Jane	
FirstName		

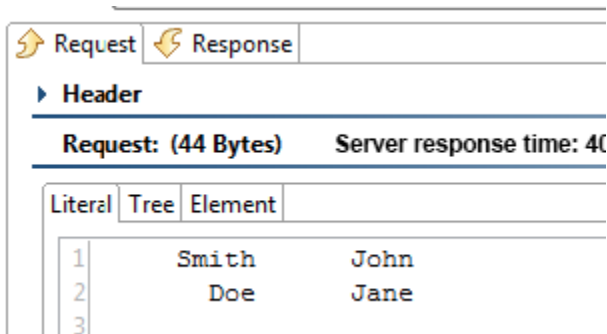
7. In the **Transport** tab, enter `http://localhost:8080/parabank/echo.jsp` in the **Router Endpoint** field.

▼ Router Endpoint

Fixed	http://localhost:8080/parabank/echo.jsp
-------	---

- 8. Save the editor.
- 9. Run the test.

10. Open the Traffic Viewer, and validate that the sent content looks like the sample message shown at the start of this tutorial.



Extending the Fixed Length Data Model with a Variable Number of Repeating Segments

Now let's assume that for each person, we want to specify a variable number of pets owned by that person. We will start with the data format we used above, but for each record we will add the following:

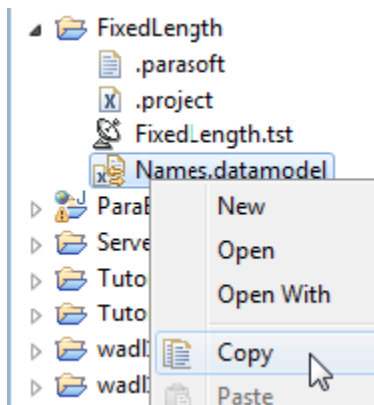
- A number field that specifies the number of pets that is 2 characters long, padded with 0.
- A variable number of sections, where each section contains the type of pet and the name of the pet. The number of sections is equivalent to the number in the field that specifies the number of pets. The field for type of pet is 5 characters long, right-aligned, and the field for the name of pet is 10 characters long, right-aligned.

Here is an example of such data:

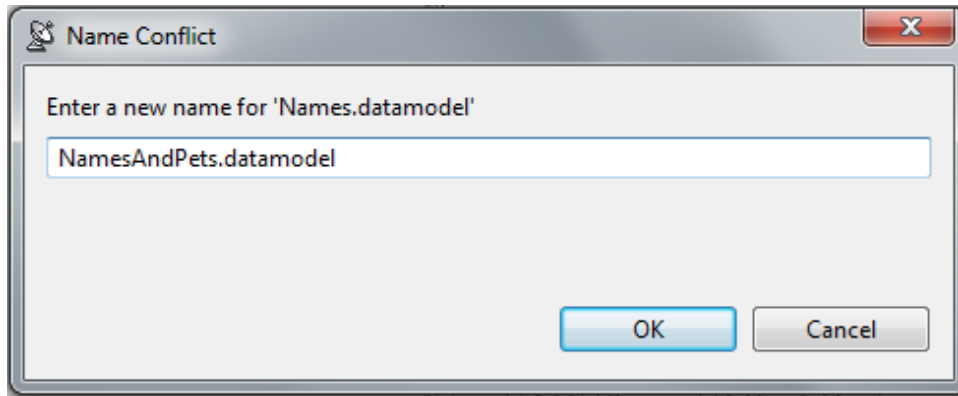
```
Smith    John00
Doe      Jane02   Cat   Fluffy Dog   Spot
```

Let's start with our existing data model, and extend it to make a new data model:

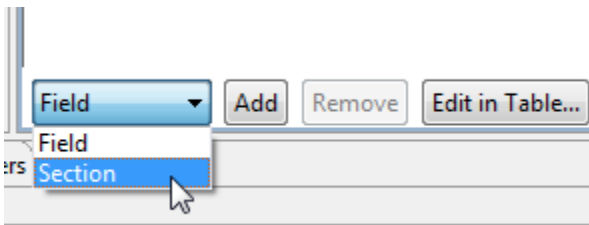
1. Create a new data model file as follows:
 - a. In the Navigator view, open the **FixedLength** project.
 - b. Right-click the project and choose **Refresh** to ensure that the view is up to date.
 - c. Right-click the **Names.datamodel** file, and choose **Copy**.



- d. Paste it into the **FixedLength** project.
- e. When the new name dialog comes up, enter `NamesAndPets.datamodel`.



2. Double-click the new **NamesAndPets.datamodel** file that is in the FixedLength project.
3. Enter `Names` and `Pets` for the name and remove the description.
4. Select **Components** and click **Add**. This adds a new field.
5. Change **Field** to **Section**, then click **Add** again. This adds a new section.



Data Model

A data model defines how to convert data to and from XML. SOAtest detects each data model stored i registered to provide data models; these models then appear for use in tools.

Format:

Name:

Description:

[-] Section	
[+] General	Name: people [...]
[-] Components	
[+] Field	Name: LastName [...]
[+] Field	Name: FirstName [...]
[+] Field	Name: fieldName [...]
[-] Section	
[+] General	Name: section [...]
[-] Components	

6. For the newly-added (third) field in the top-level section, expand the node for the field and specify the following:
 - **Name** - NumberOfPets
 - **Type** - integer
 - **Length** - 2
 - **Alignment** - right
 - **Padding** - 0

[-] Section	
[+] General	Name: people [...]
[-] Components	
[+] Field	Name: LastName [...]
[+] Field	Name: FirstName [...]
[-] Field	
	Name
	Type
	Length
	Alignment
	Padding
	NumberOfPets
	integer
	2
	right
	0
[-] Section	
[+] General	Name: section [...]

- For the new section, expand the **General** node and specify the following:
 - Name** - pets
 - Recordname** - pet
 - Criteriafornextrecord** - xpath
 - XPath** - `count($section/*) < $section/../NumberOfPets`
- Select the **Components** node in the new section, ensure **Field** is selected, and click **Add** twice.

General	
Name	pets
Record name	pet
Record separator	
Criteria for next record	xpath
XPath	<code>count(\$section/*) < \$section/../NumberOfPets</code>

- Edit the components as follows:
 - Select the **Components** node in the new section, then click **Edit in Table**.

- In row 1, specify `PetType` for **Name**, 5 for **Length**, and `right` for **Alignment**.
- In row 2, specify `PetName` for **Name** and `right` for **Alignment**.

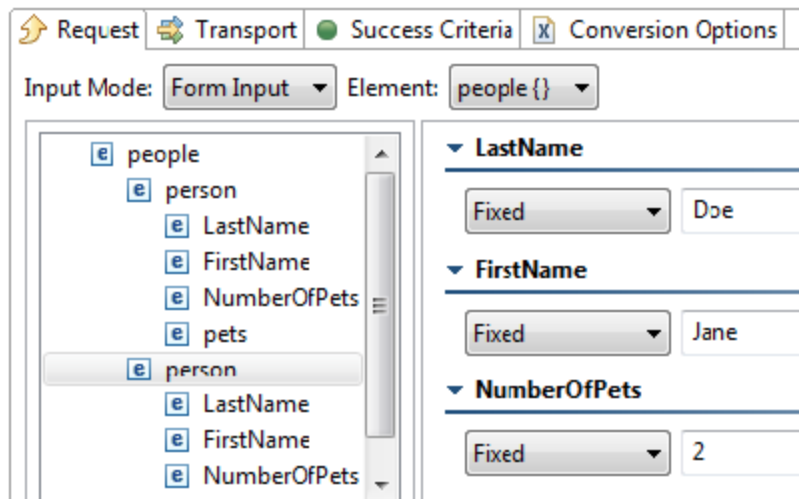
	Name	Type	Length	Alignment
1	PetType	string	5	right
2	PetName	string	10	right

- Click **Return to Full Data Model**.
 - Save the editor.
- Add a new Fixed Length Client to the Test Suite of the FixedLength.tst file and select **Names and Pets**: as the **Message type**.

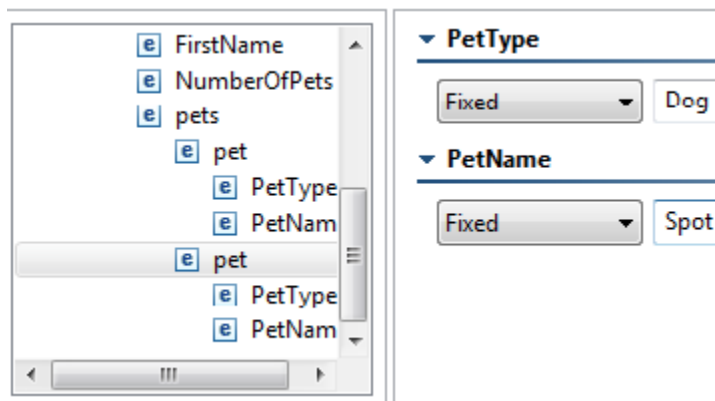
11. Enter the first person's information as follows:
 - a. In the **Request** tab, select the **people** element.
 - b. Modify the default values as follows:
 - **LastName** - Smith
 - **FirstName** - John
 - **NumberOfPets** - 0

- c. Delete the **pets** element.

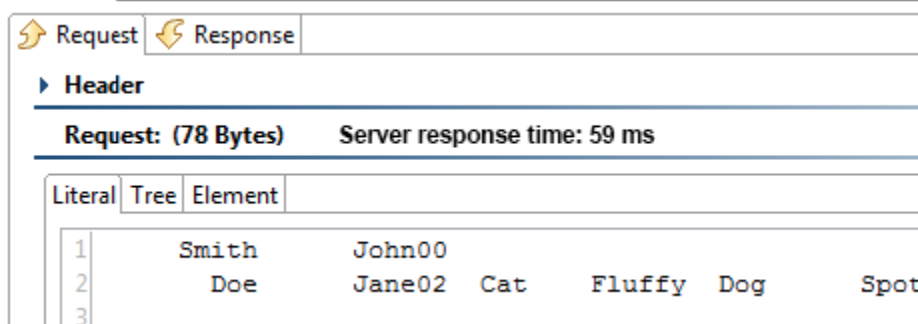
12. Enter the second person's information as follows:
 - a. Add a second **person** element by selecting the **people** element, then clicking the **Insert** button below the table.
 - b. Enter the following values:
 - **LastName** - Doe
 - **FirstName** - Jane
 - **NumberOfPets** - 2



- c. Right-click the **pets** node for the second person, choose **Insert Multiple**, specify 2 for the **Count**, then click **OK**.
- d. For the first pet, specify the following:
 - **PetType** - Cat
 - **PetName** - Fluffy
- e. For the second pet, specify the following:
 - **PetType** - Dog
 - **PetName** - Spot



13. In the **Transport** tab, set the **RouterEndpoint** to `http://localhost:8080/parabank/echo.jsp`.
14. Save the tool.
15. Run the test.
16. Open the Traffic Viewer, and validate that the content that was sent appears like the example data we are modeling.



The .tst file corresponding to these exercises can be found at `<SOAtest install dir>/examples/tests/FixedLength.tst`.