IBM Integration Bus

Message Modeling with DFDL

Lab 3
Record-oriented, tagged, delimited text

June 2015
Hands-on lab built at product
Version 10.0.0.0

1. INTRODUCTION 3

1.1 LAB PREPARATION 3
1.2 LAB SCENARIO 3

2. BUILD THE MESSAGE MODEL 4

3. REFINING THE MESSAGE MODEL 11

4. TESTING THE MESSAGE MODEL 42

END OF LAB GUIDE 48

1. Introduction

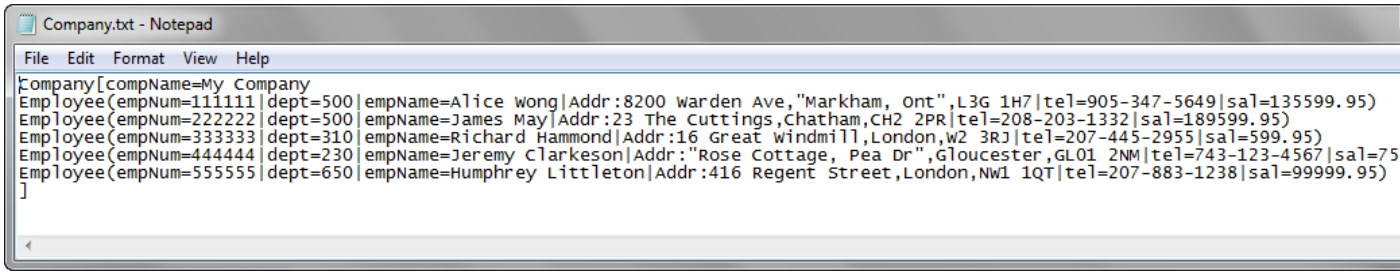
1.1 Lab preparation

To run this lab, unzip the supplied file MessageModelling.zip into the directory c:\student10 directory. This will create a subdirectory called MessageModelling, with several further subdirectories. If you are using the pre-supplied vmware image, this will already be available.

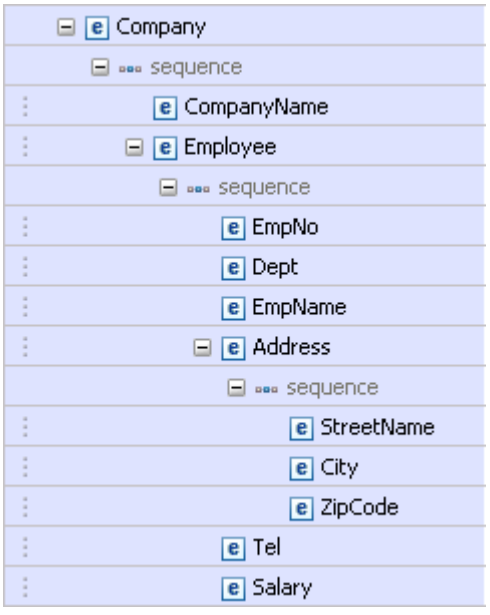
1.2 Lab Scenario

A Record oriented message model is useful to model messages that consist of text strings, but it can also handle binary data. Examples of this type of messages are those that conform to the ACORD AL3, EDIFACT, HL7, SWIFT, or X12 standards. This format allows a high degree of flexibility when defining message formats, and is not restricted to modeling specific industry standards, so you can use it to model your own messages.

In this lab you will build a message model capable of parsing this tagged / delimited file:



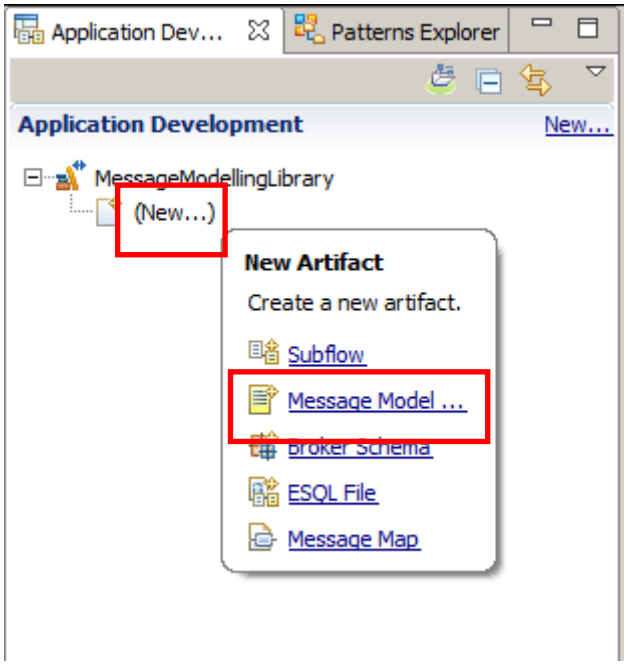
This is an outline of the final message model that you will create:



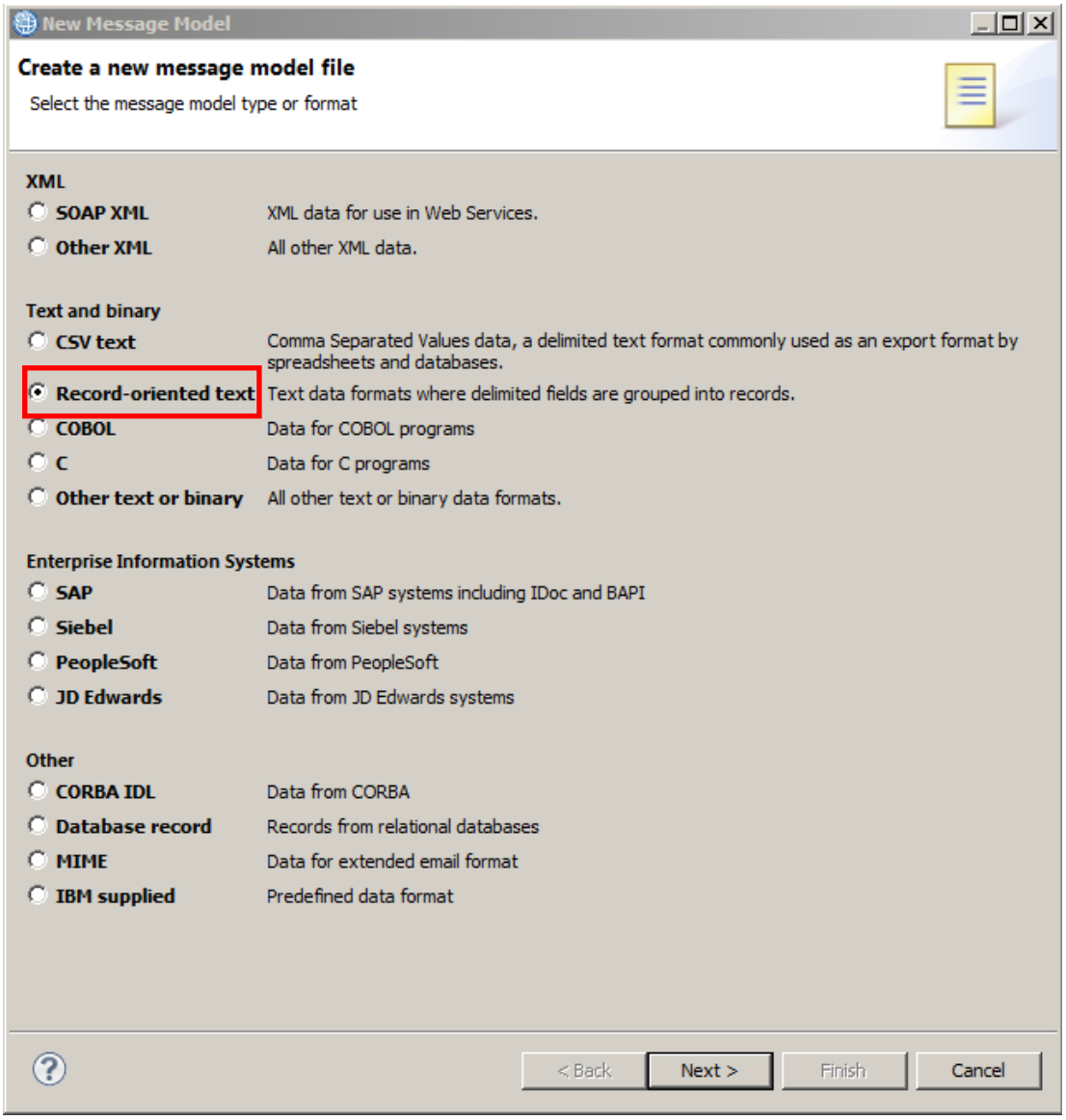
2. Build the Message Model

1. In the MessageModellingLibrary that you created in Lab1, click New -> Message Model.

   (If you didn't do any earlier labs, you can create a new library called MessageModellingLibrary).

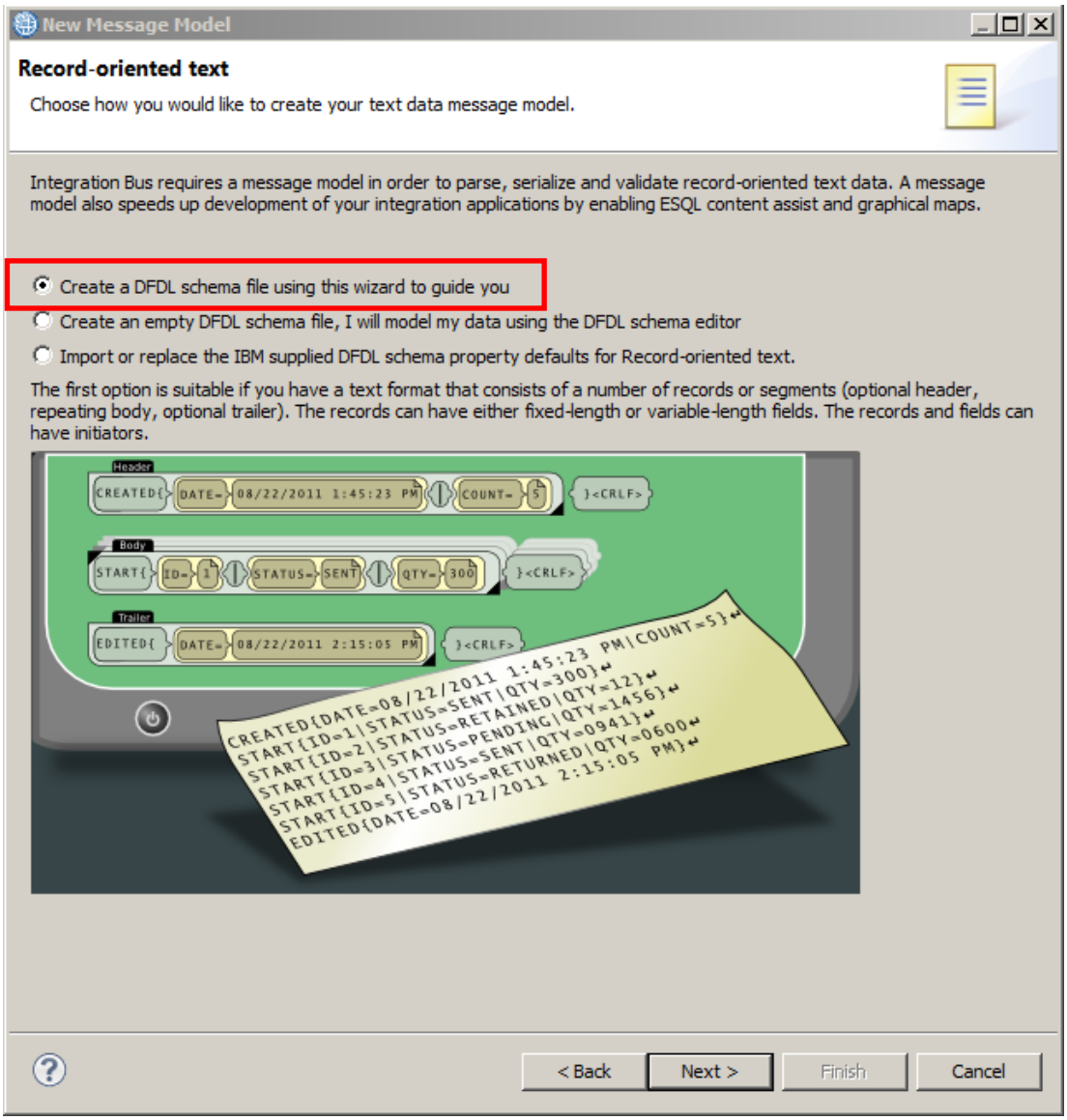


2. In the "New Message Model" window, select "Record-oriented text" and click Next.

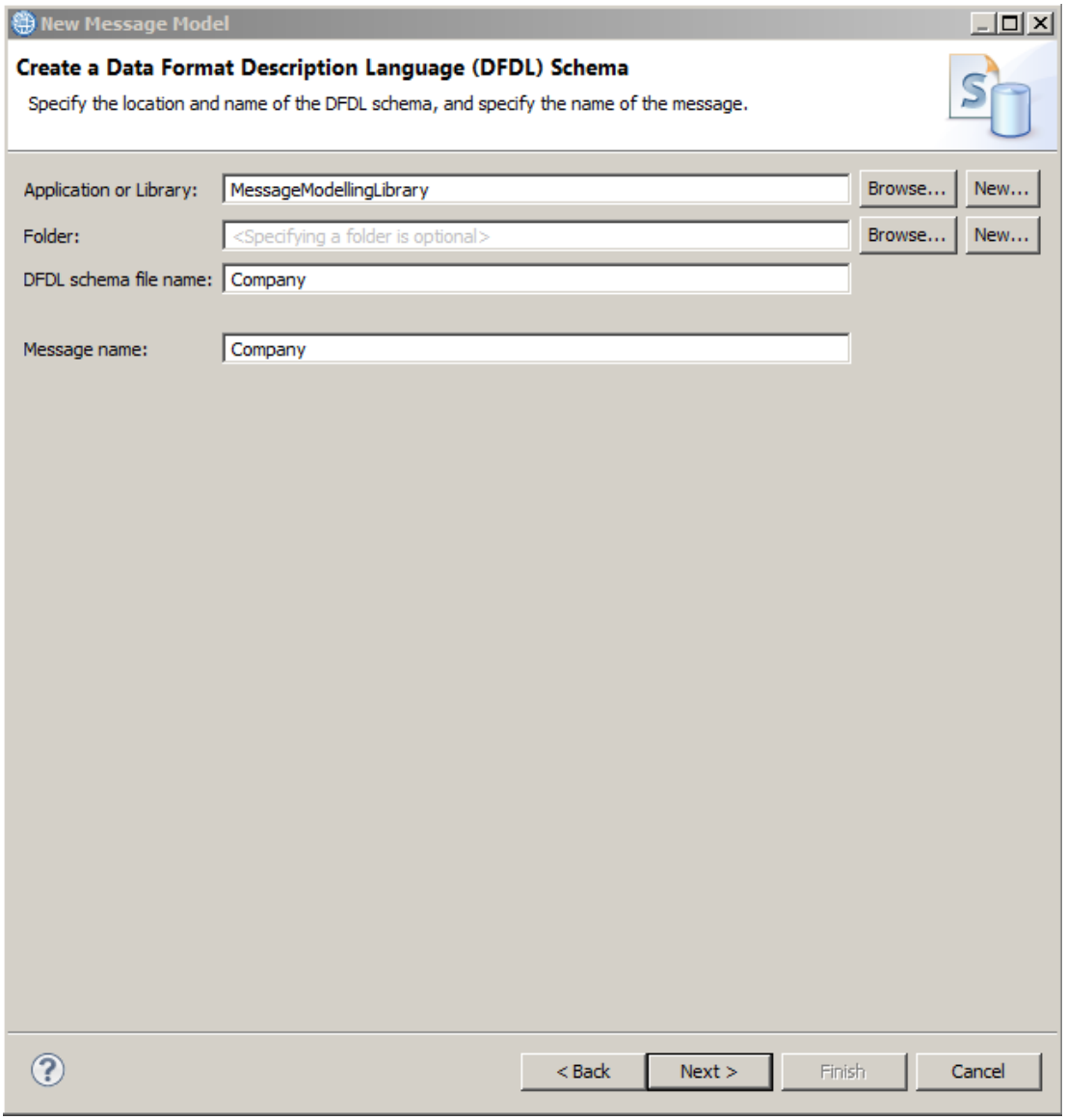


3. You can create the new message model using a wizard or create an empty DFDL schema and start from scratch.

   Leave the default selection to “use the wizard” and click Next.



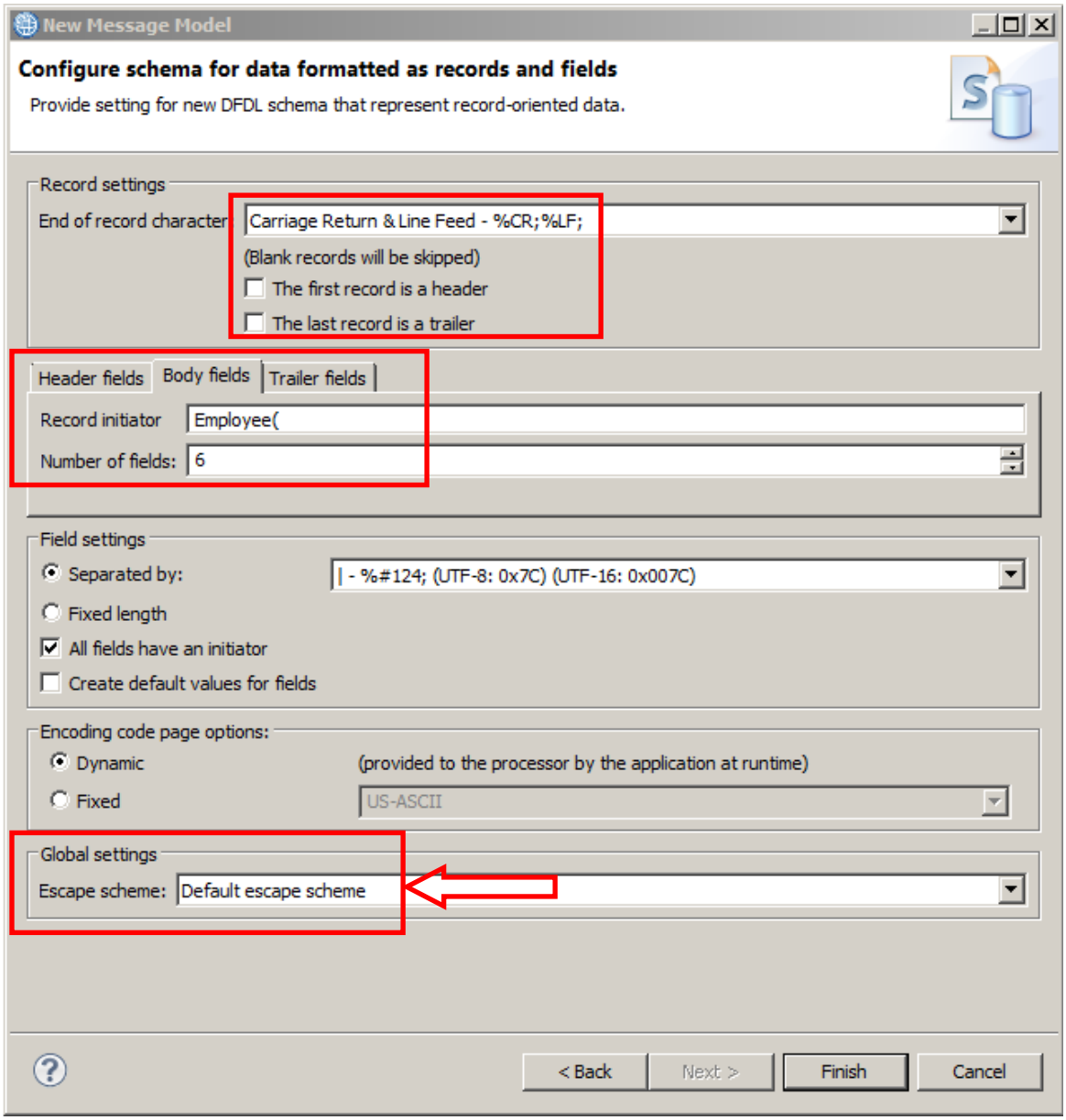
4. Enter "Company" as the name for the DFDL Schema and click Next.



5. Uncheck both "The first record is a header" and "The last record is a trailer".

   On the "Body fields" tab, set the Record initiator to “Employee(" and set the number of fields to 6.

   Change the Escape scheme to "**Default escape scheme**". Note that in versions of IIB prior to V9.0.0.2, the Escape scheme was automatically set to this value. The default escape scheme is required in this lab, because there is an element in the test data which has a value containing embedded comma (,) which needs to be escaped.

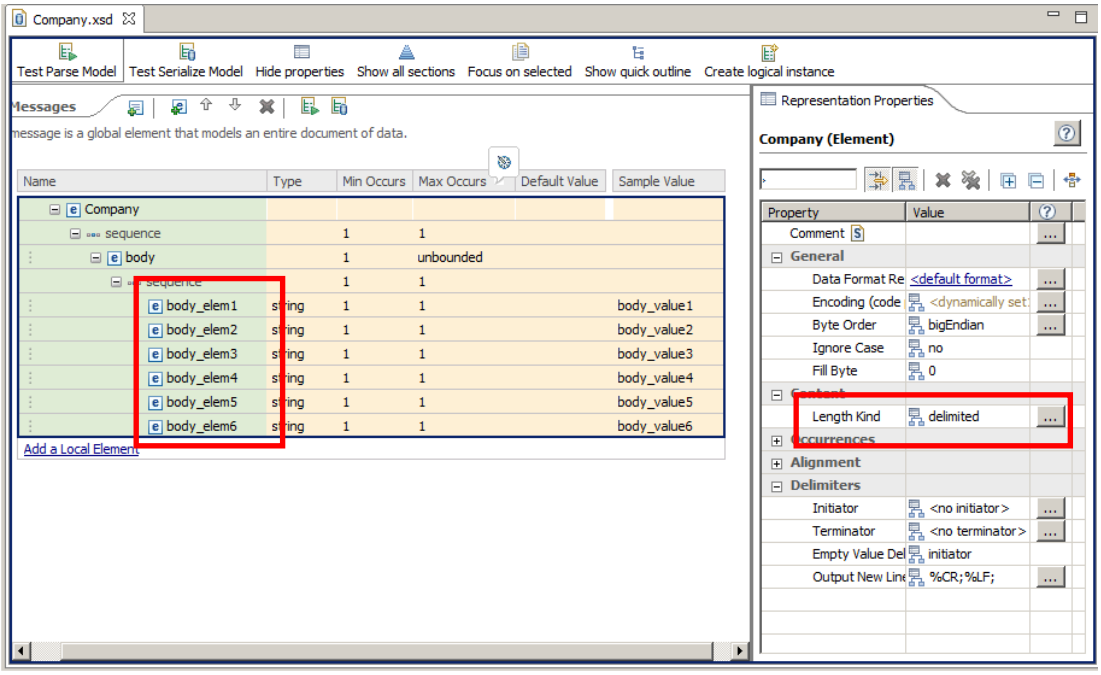


   Leave "Separated by" as "|" (pipe) and "All fields have an initiator" checked. These default values match the required for the message model.

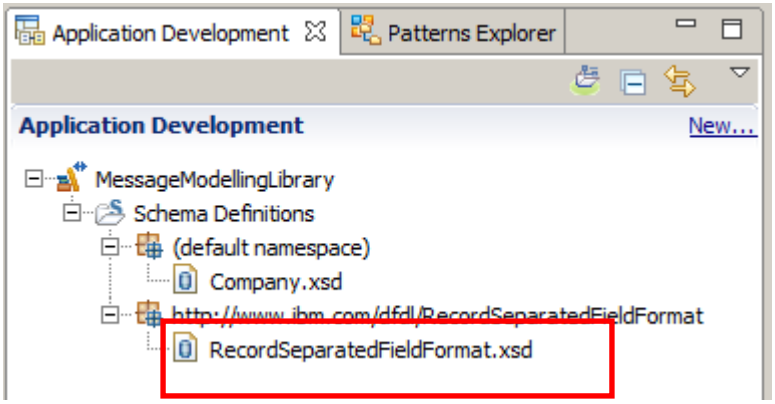
   Click Finish.

6. When the wizard finishes, the DFDL Editor will open with the generated Company.xsd schema file.

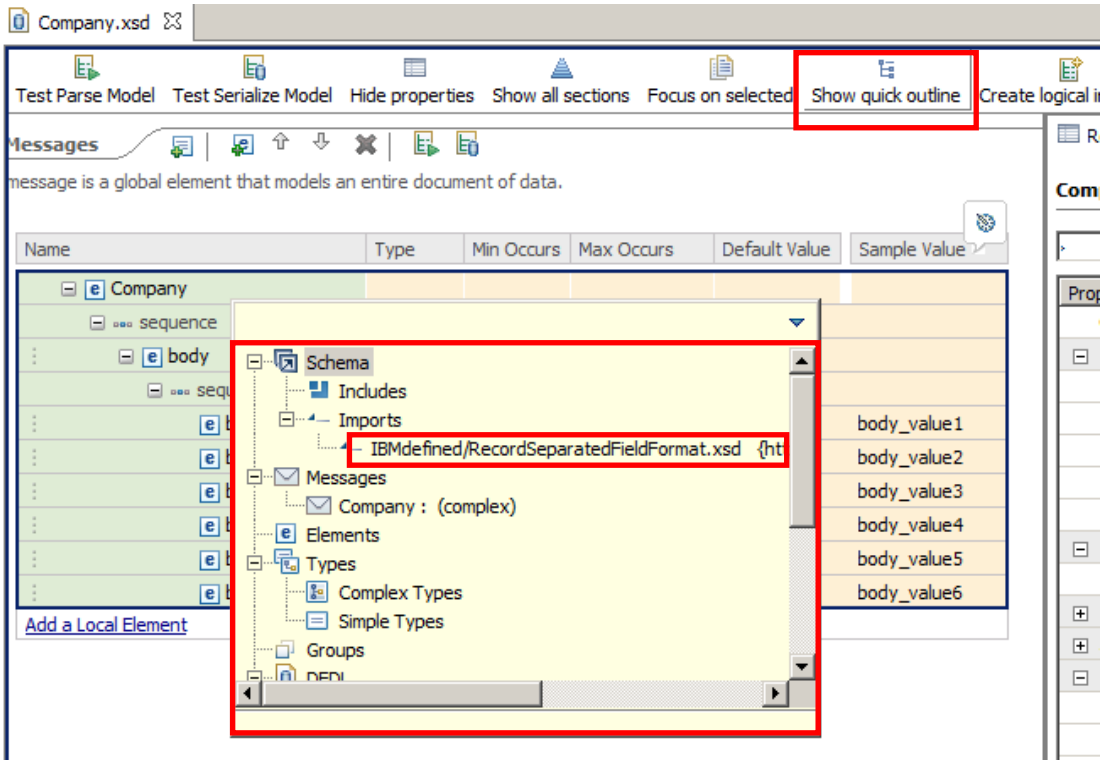
   Note that as you defined it in the wizard, the model has six fields, "Length Kind" property is set to delimited.



7. The wizard has also generated a second xsd file, RecordSeparatedFieldFormat.xsd. This is the "Helper schema file" that contains the default values for all DFDL properties for the defined Record Oriented data format.



8. Click on the "Show quick outline" icon.

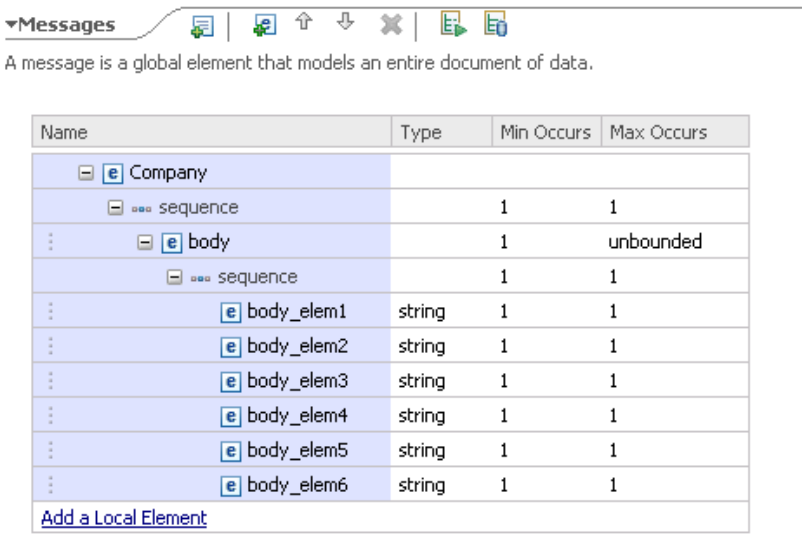


The Outline view will appear with a high level view of the elements of your message model. If you click on any of them, the editor will focus on it.

In the outline view you can see that Company.xsd has a reference to the helper schema file RecordSeparatedFieldFormat.xsd.

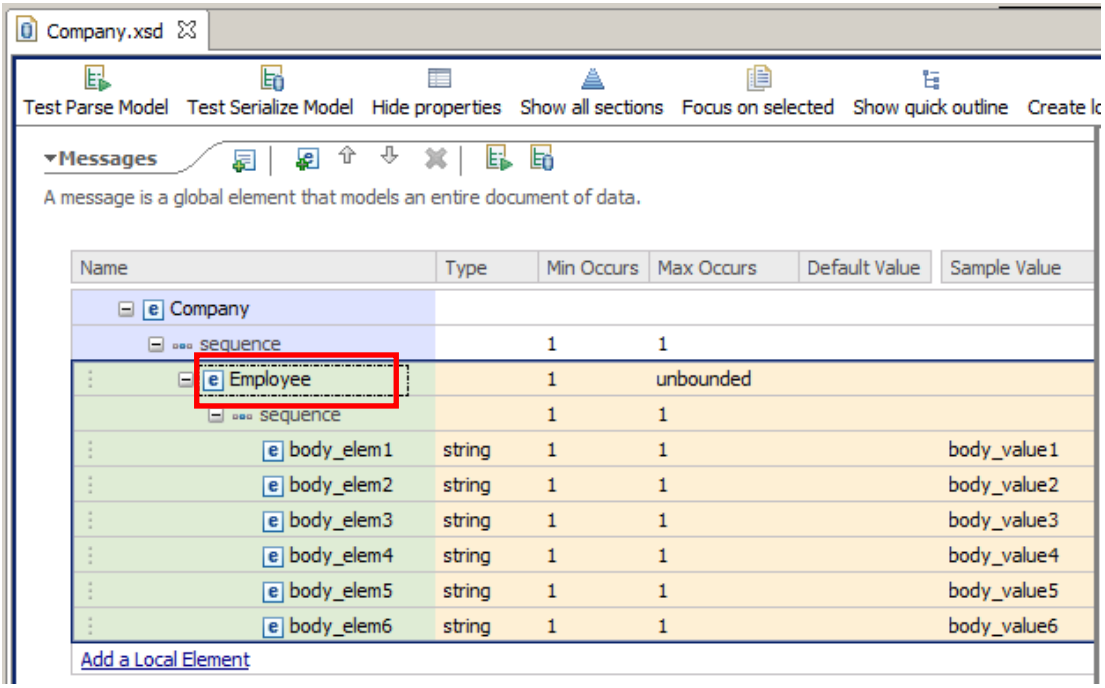
To close the Outline pop-up, click anywhere else on the editor window.

Hint: if the Messages display "disappears", click Show all Sections (the blue pyramid), and then expand Messages, then expand “body”. You can optionally click “Hide Empty Sections” to provide a less cluttered display.

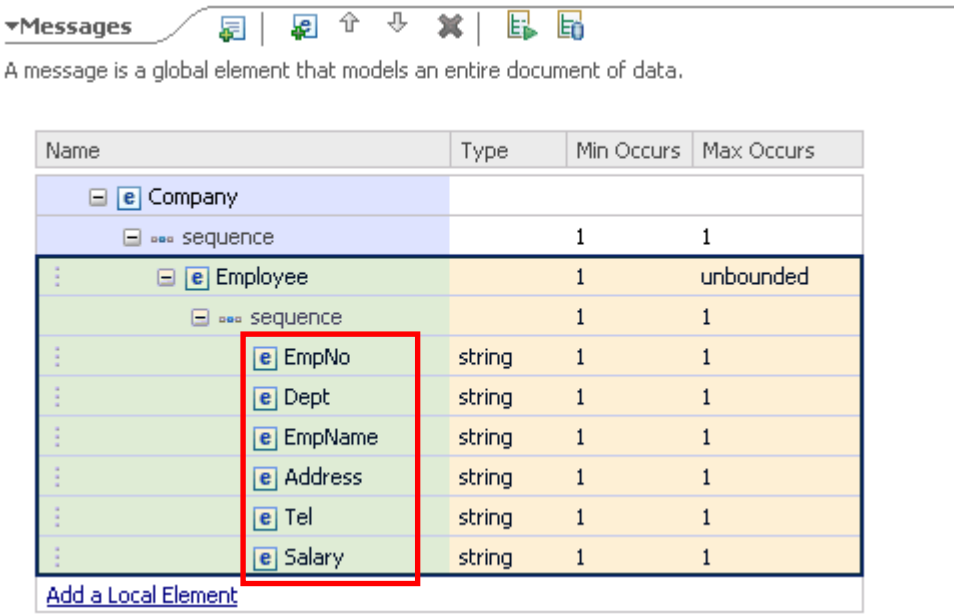


3. Refining the Message Model

1. Change the name of the "body" field to "Employee" by single-clicking on it, and overtyping.

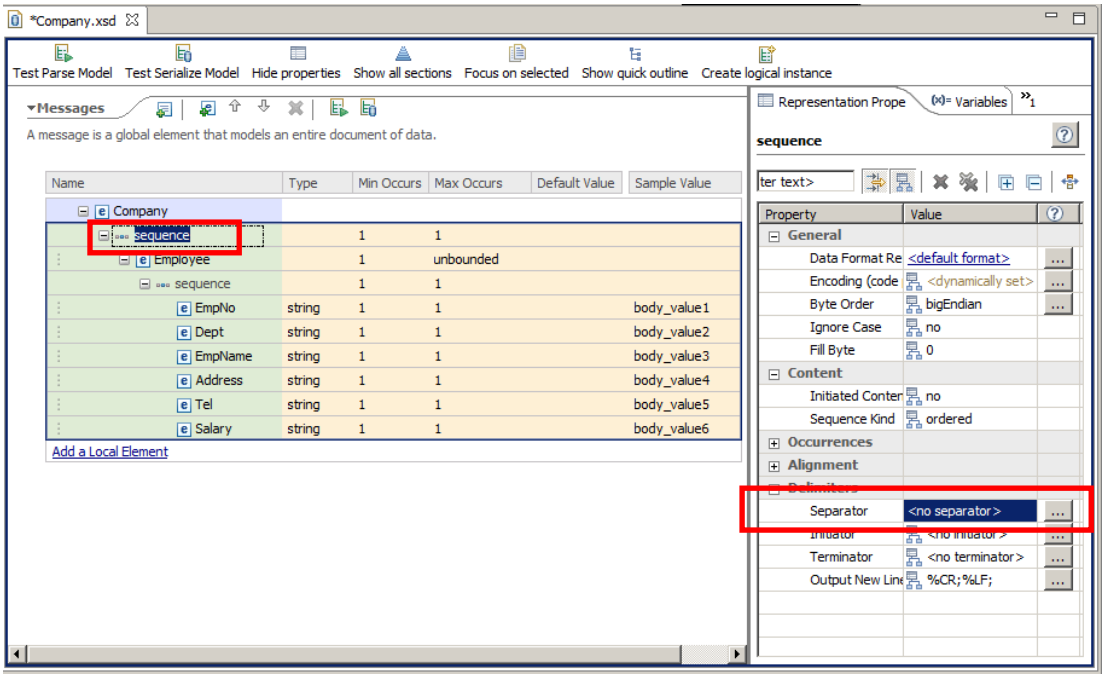


2. Similarly, change the name of the 6 fields under "Employee" as shown. You can just use the down-arrow to move between element names.

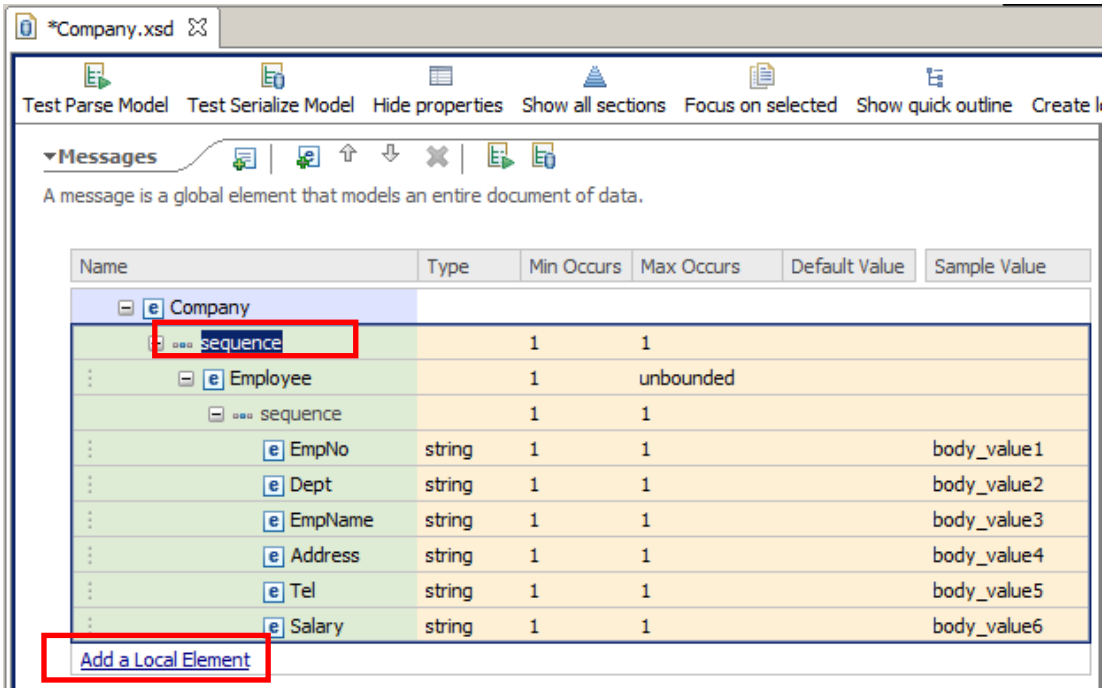


3. Click on the <sequence> content of the Company element. In the Representation Properties view, go to the "Delimiter" section and delete the value of the Separator field (%CR;%LF;%WSP*).

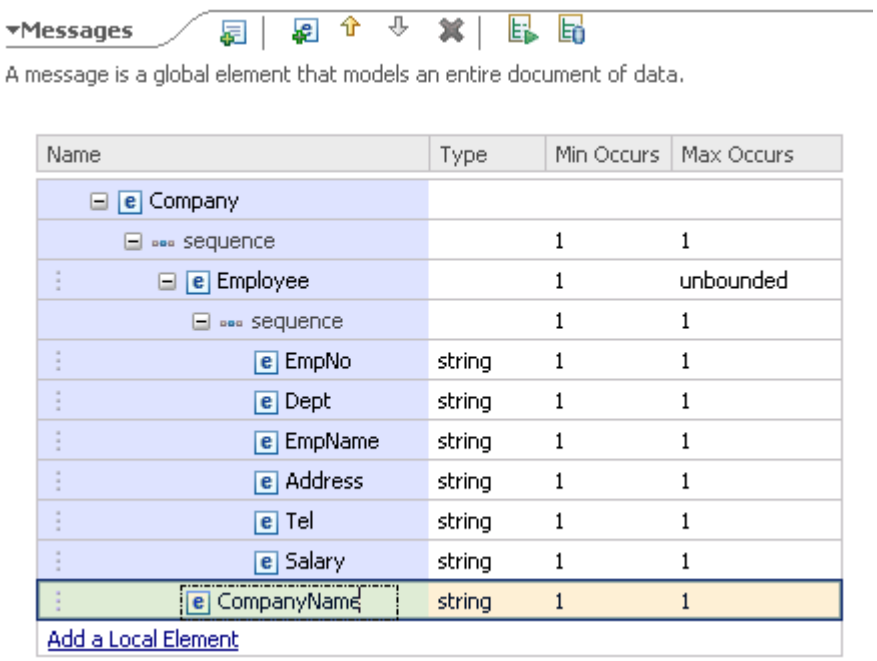
   This value was introduced by the wizard, but since it doesn't comply with our data file, we need to delete it. (Highlight the value, click delete, and then click return to make sure the change takes effect).



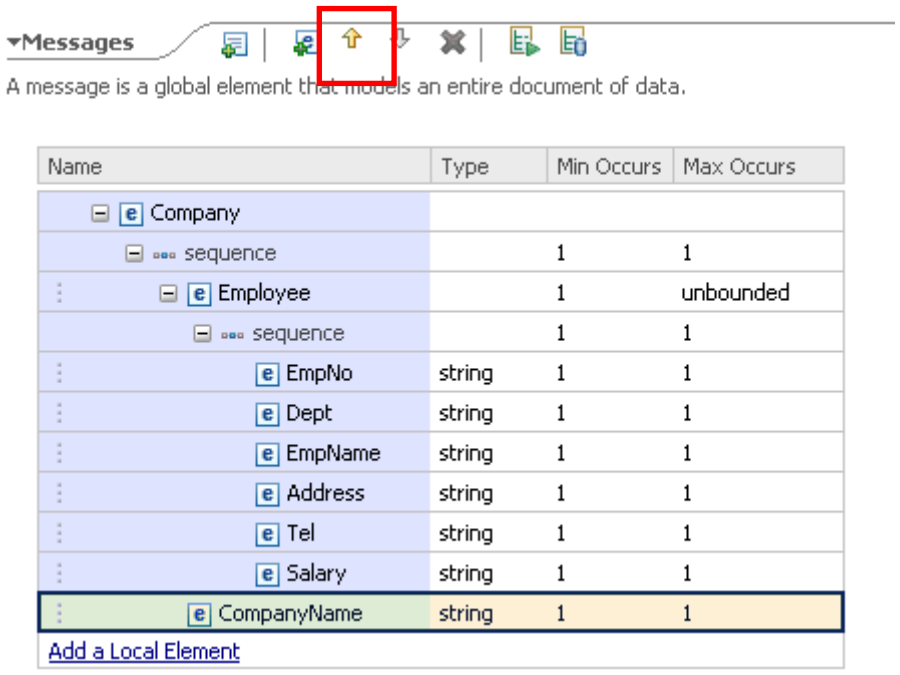
4. Click on the <sequence> content of the Company element and click on "Add a Local Element".



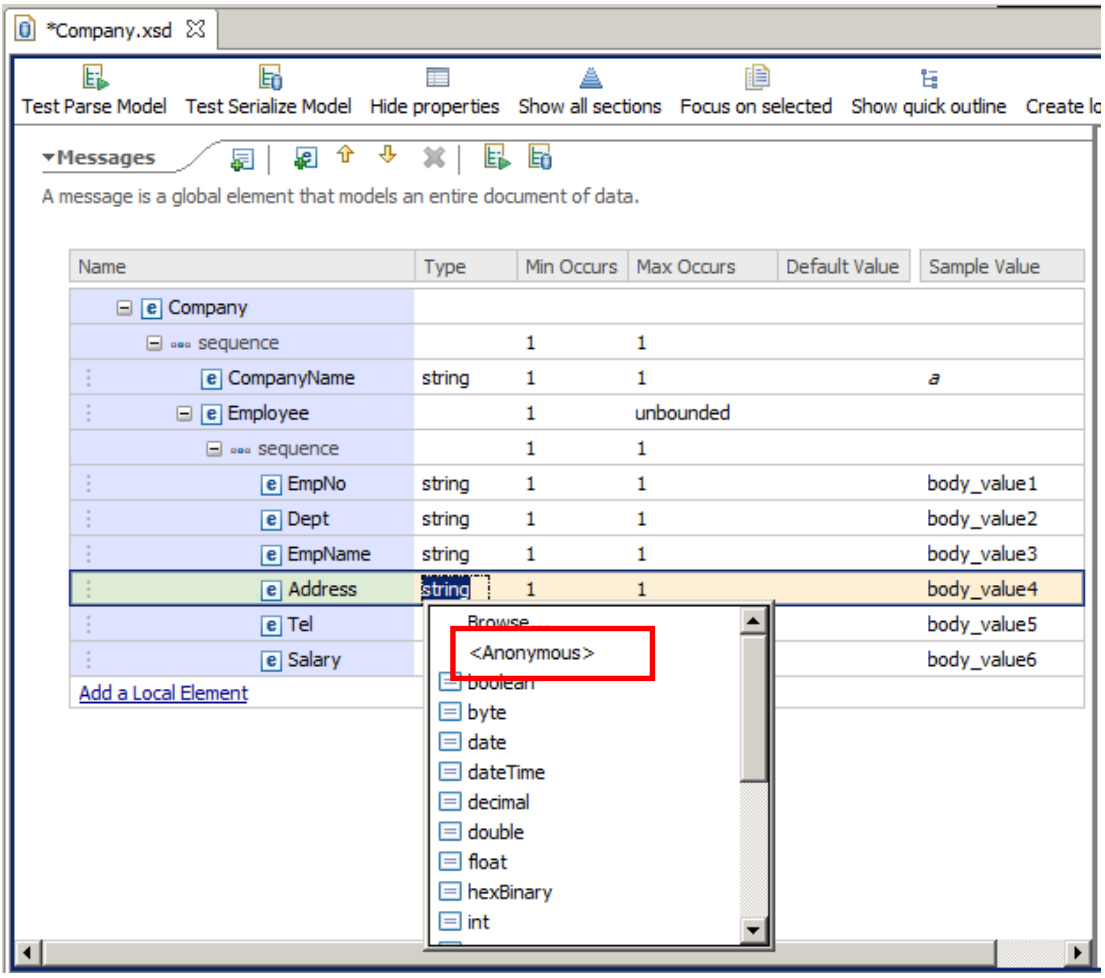
5. Name the new element "CompanyName"



6. Highlight "CompanyName" and click the yellow "Up" arrow to move this element above the "Employee" element (or you can right-click the element and select “Move Up”.)

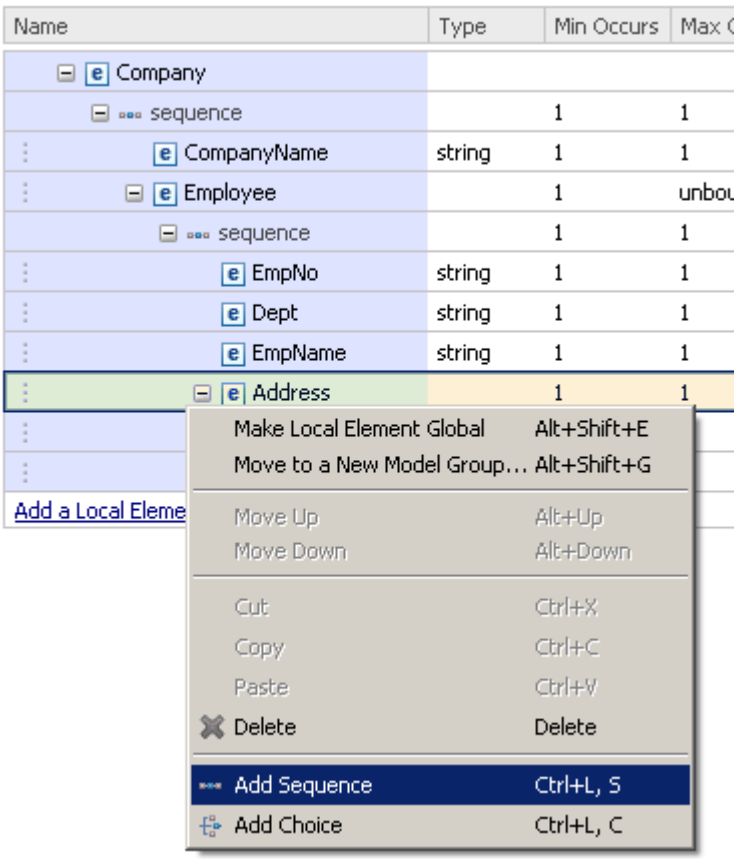


7. Click on the type column of the Address element and select "<Anonymous>"



8. Now define three elements under the Address element.

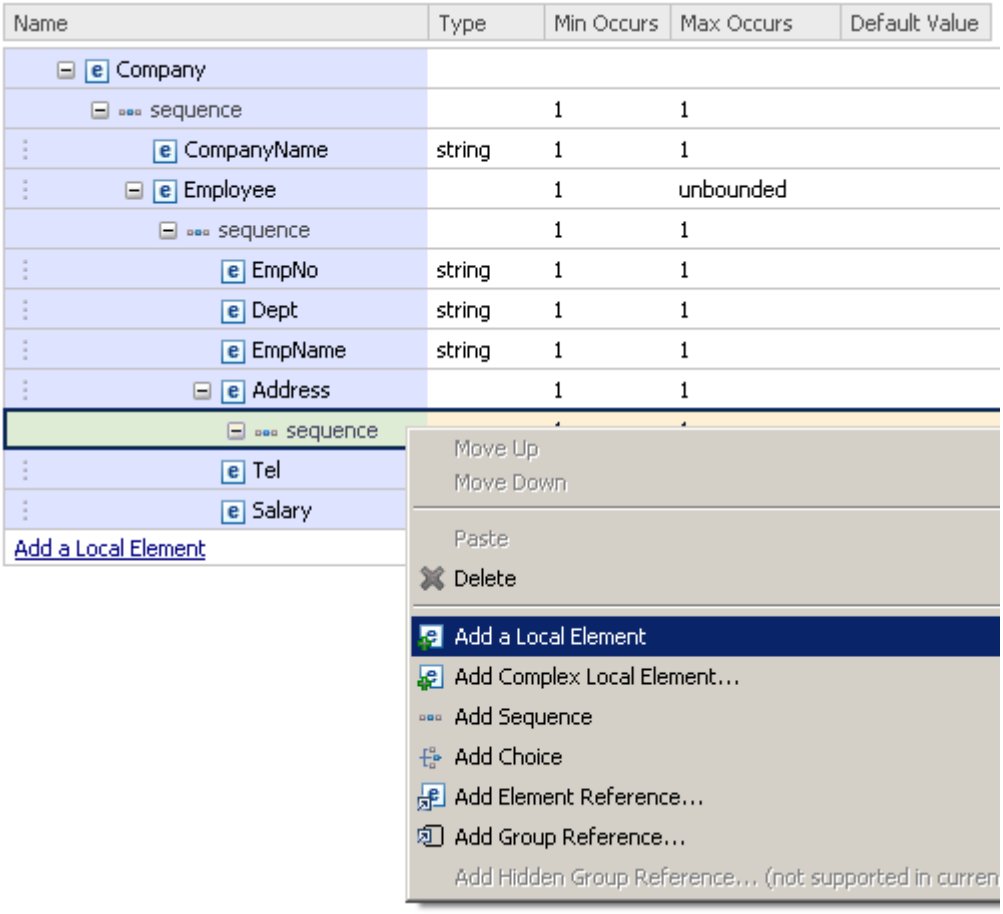
   First, you need to add a new sequence element under Address.
   Right-click on the Address element line (although not on any text) and select "Add Sequence".



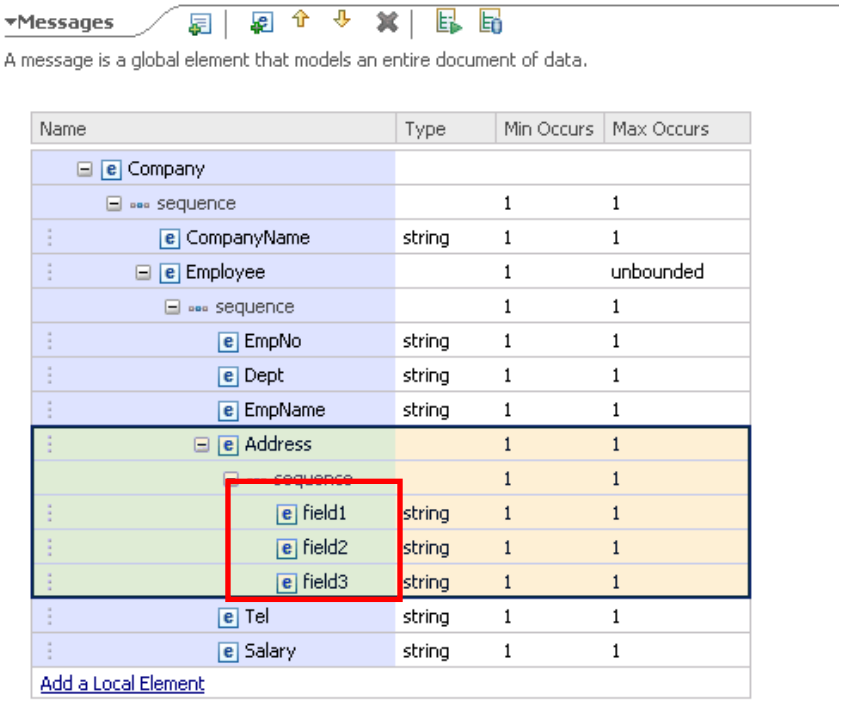
   This will be added like this:

Name	Type	Min Occurs	Max Occurs
Company			
sequence		1	1
CompanyName	string	1	1
Employee		1	unbounded
sequence		1	1
EmpNo	string	1	1
Dept	string	1	1
EmpName	string	1	1
Address		1	1
sequence		1	1
Tel	string	1	1
Salary	string	1	1
Add a Local Element			

9. Now add a new element under the new Sequence. Right-click on the Sequence line (although not the text *** sequence *** itself), and select Add a Local Element.

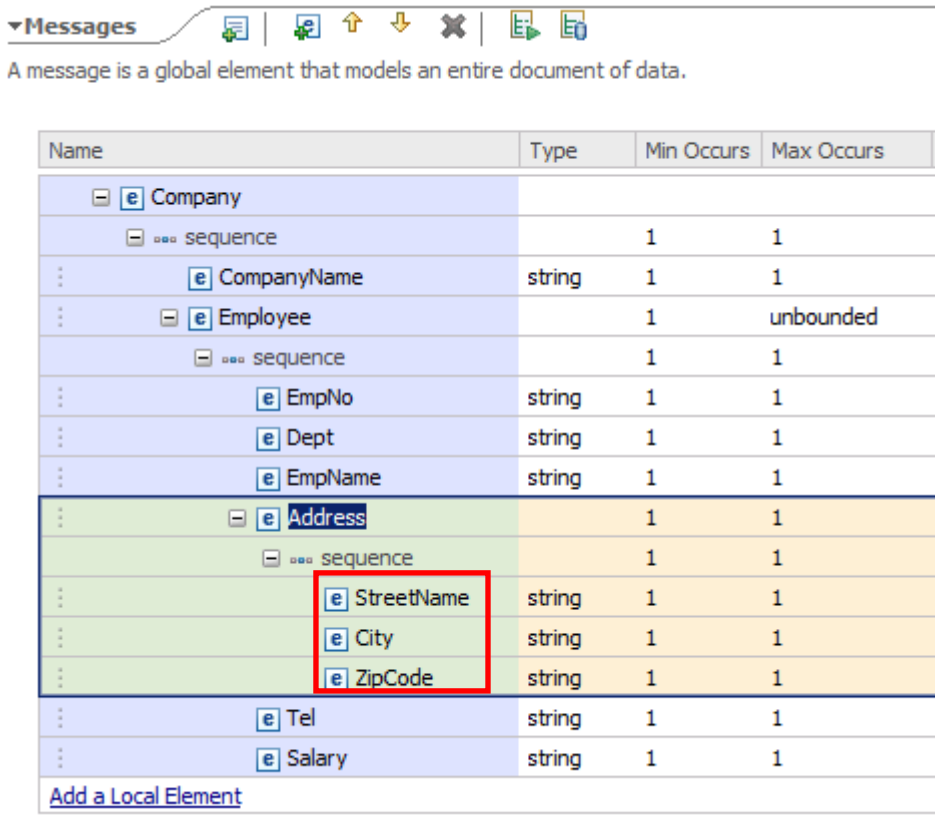


10. Repeat the previous step twice to add two more fields to the Address element.



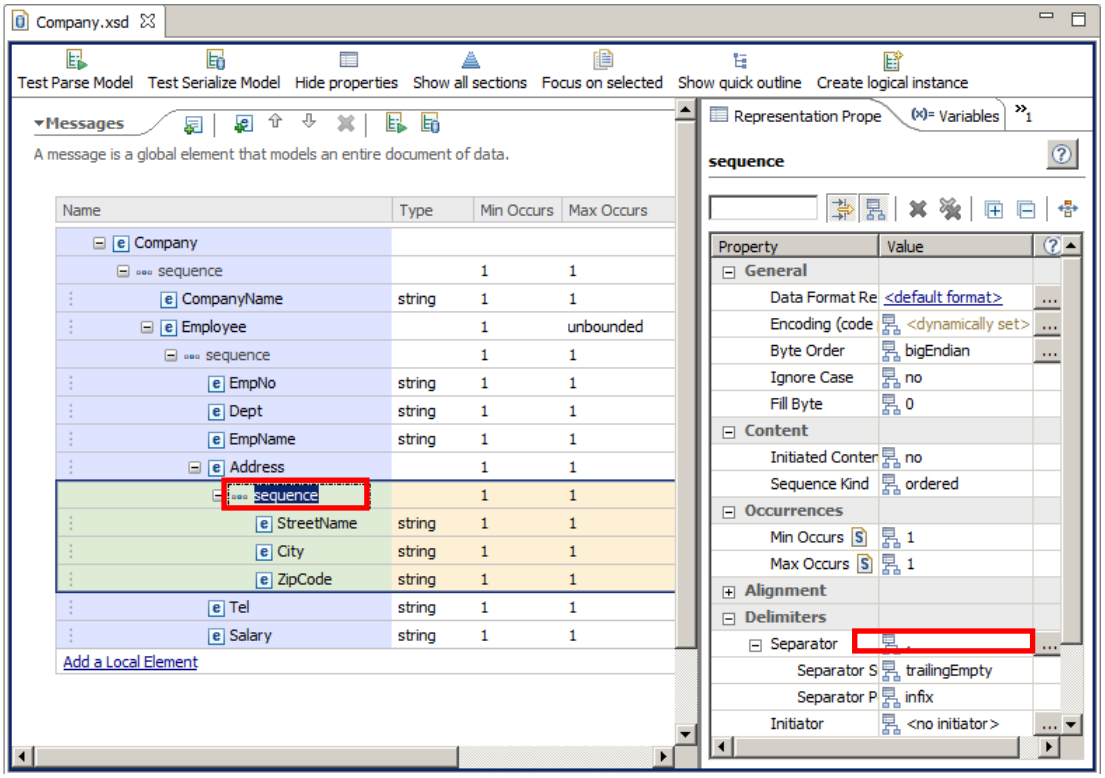
11. Change the names of the 3 elements you've just added to the following by clicking and overwriting with the new names:

    1. StreetName
    2. City
    3. ZipCode

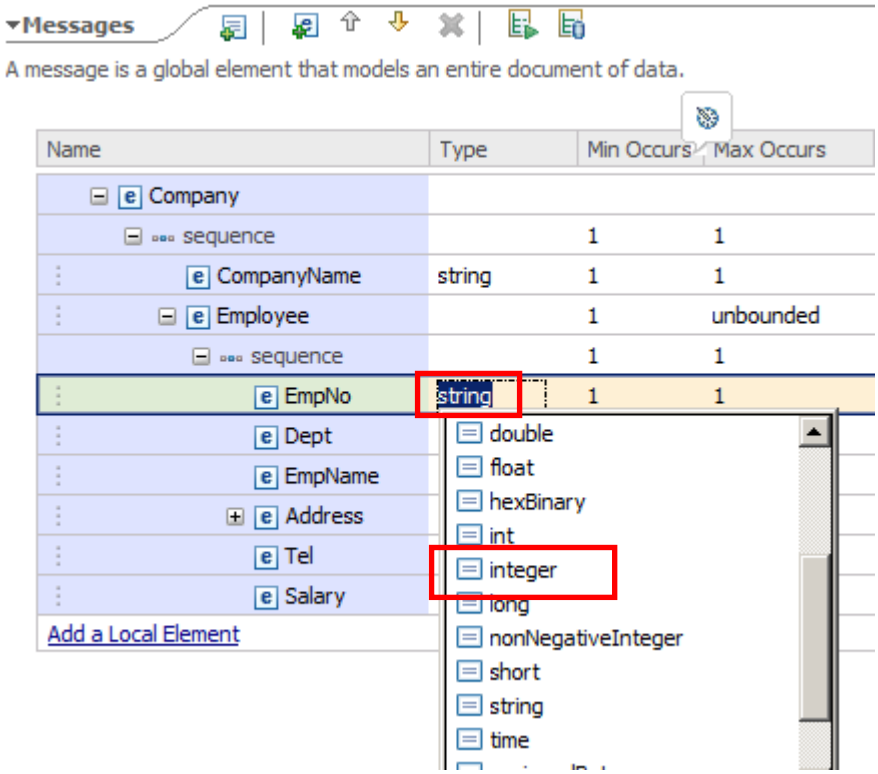


12. Click on the <sequence> content of the Address element and take a look at the Delimiters section in the Representation properties.

    Notice the inheritance icon next to the Separator field. The Separator for this element was automatically set to "," (comma) because it was inherited from the Helper Schema file (RecordSeparatedFieldFormat.xsd).

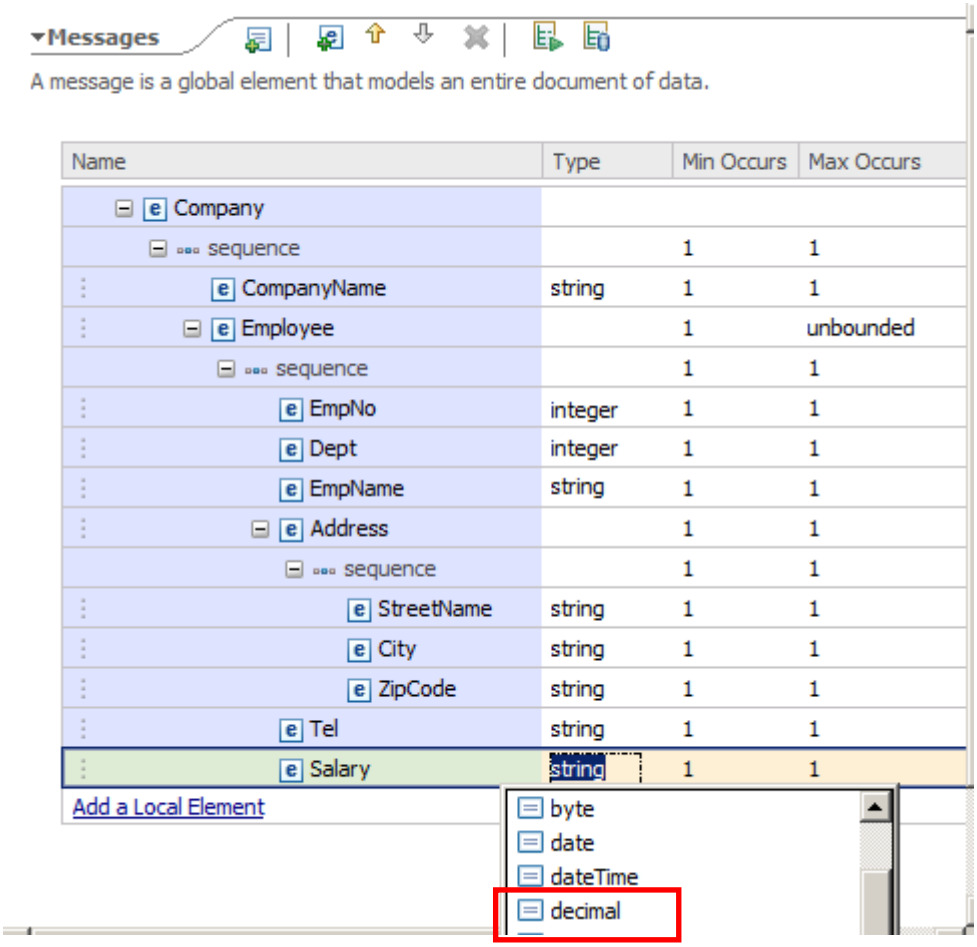


13. Click on the type column of the "EmpNo" element and select "integer" (not “int”).

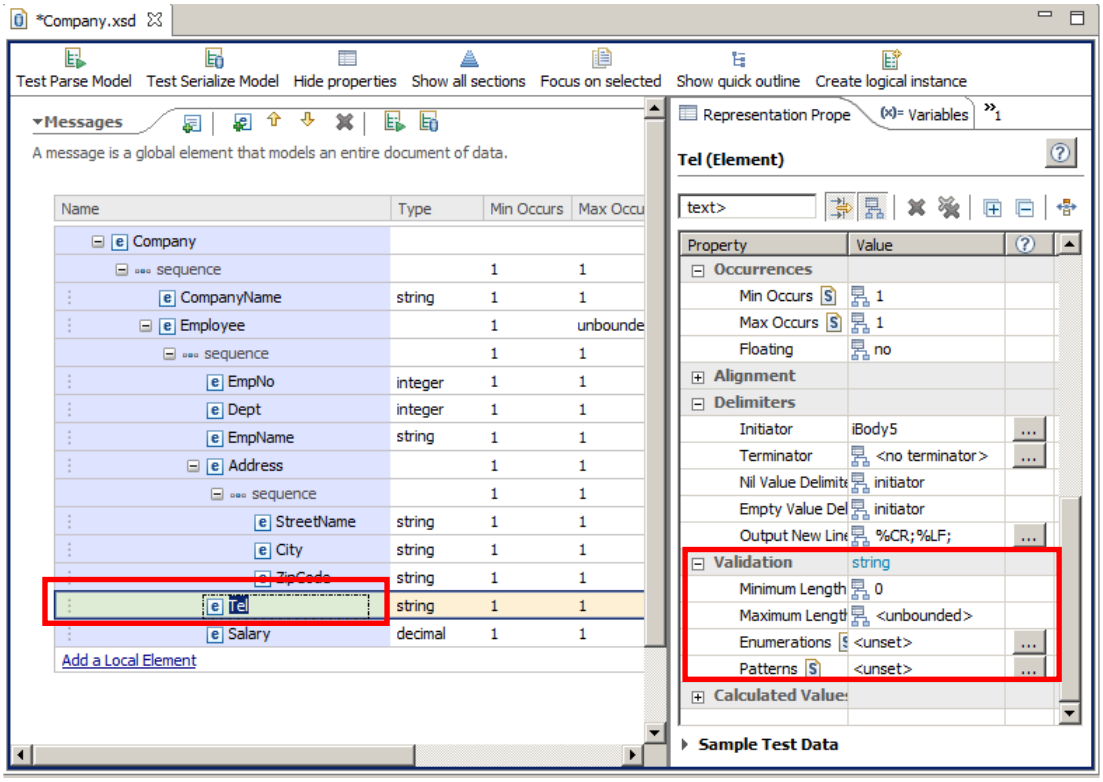


14. Similarly, set the Type of the "Dept" element. = integer (not “int”, which would restrict the value to 4 bytes).

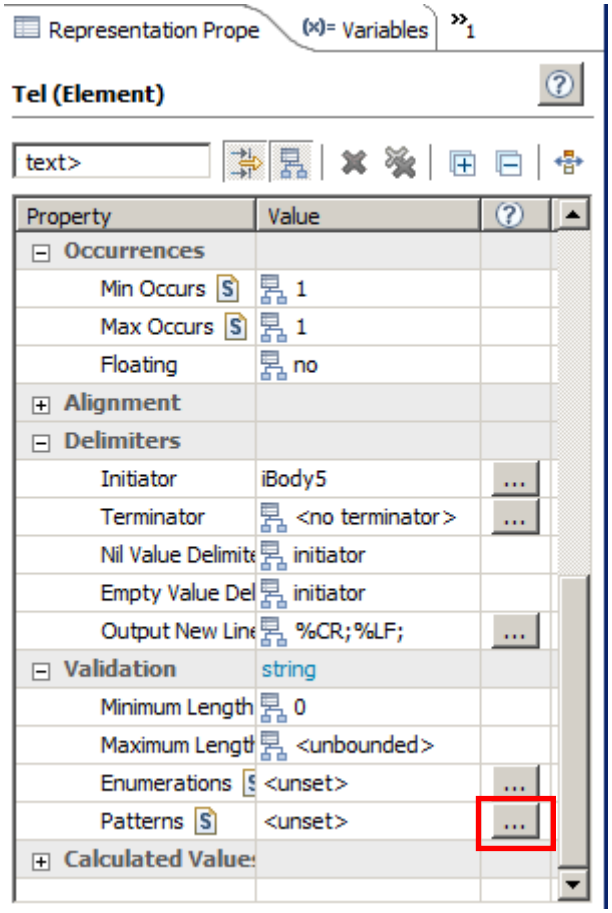
15. Set the Type of “Salary” = decimal.



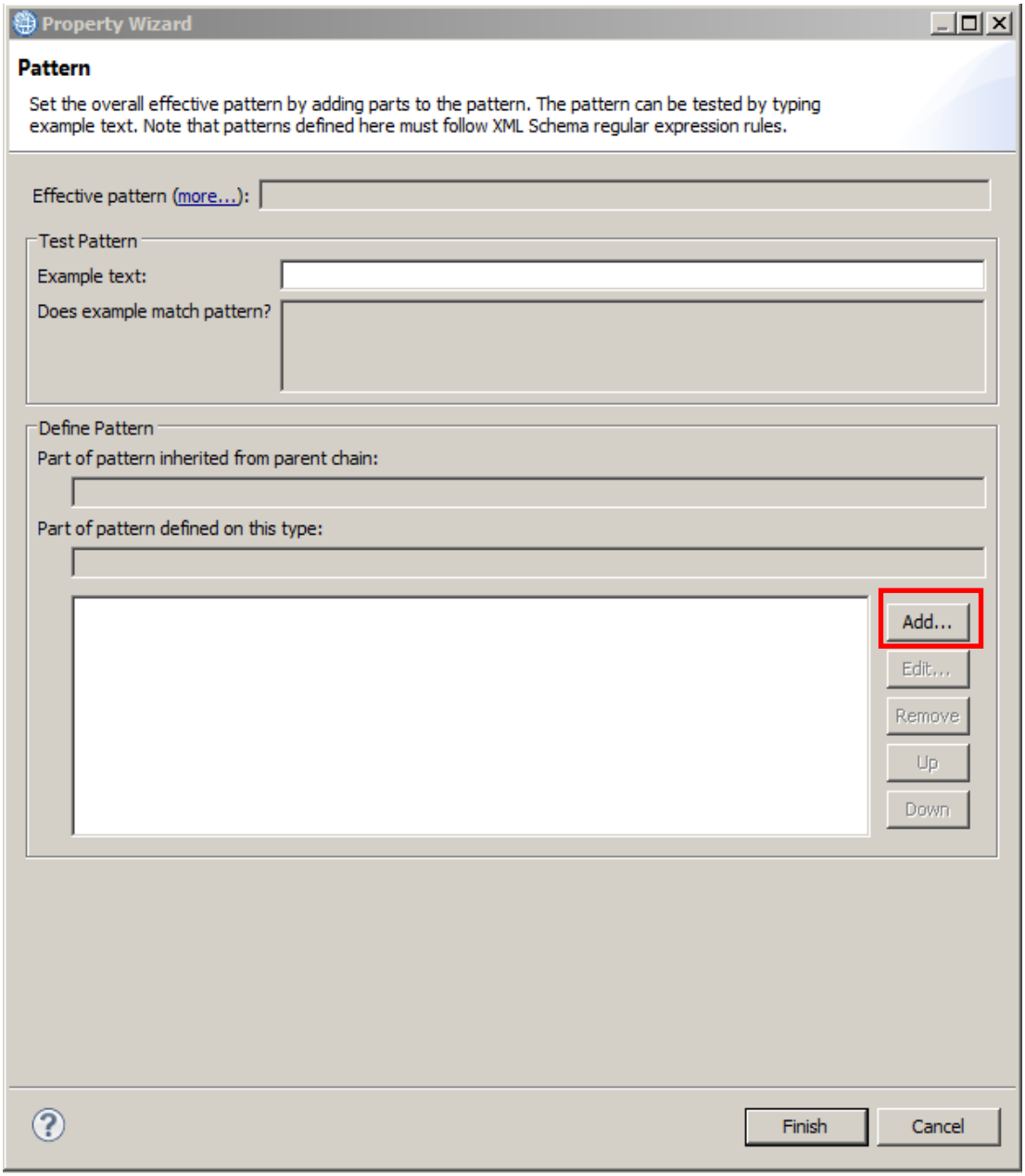
16. Highlight the "Tel" element and look for the "Validation" section in the Representation Properties view of the DFDL Editor.



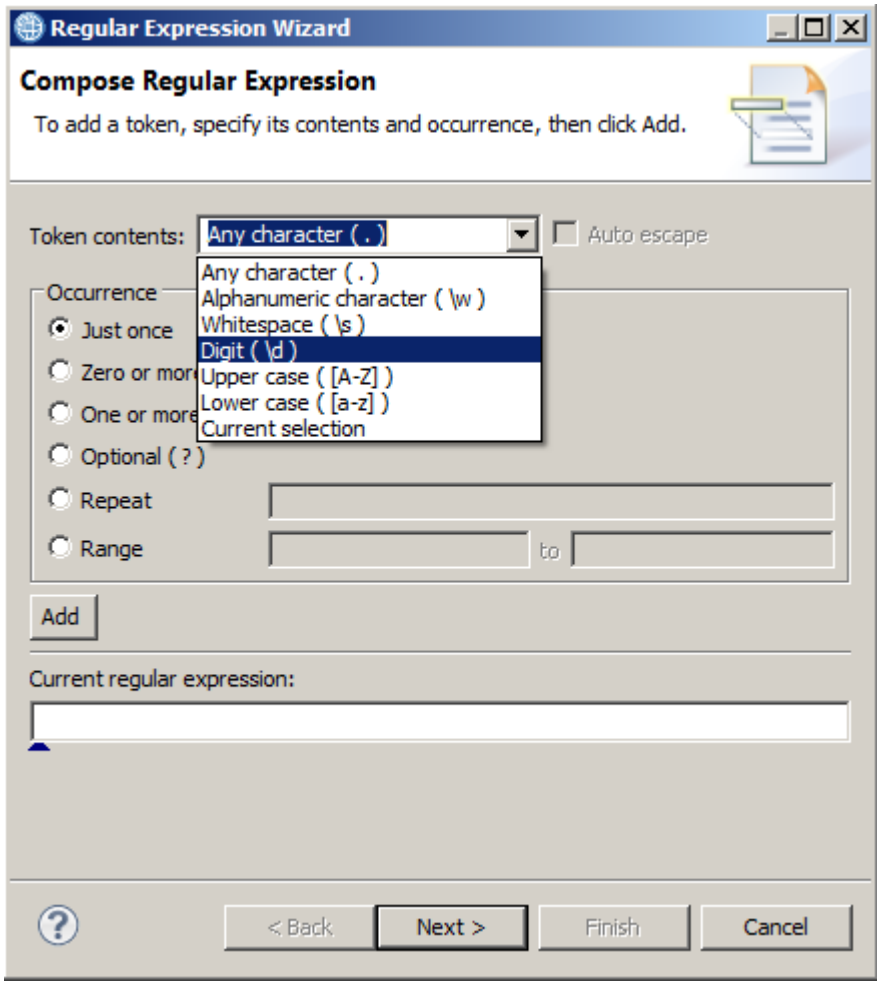
17. Click on the "..." button next to the "Patterns" property.



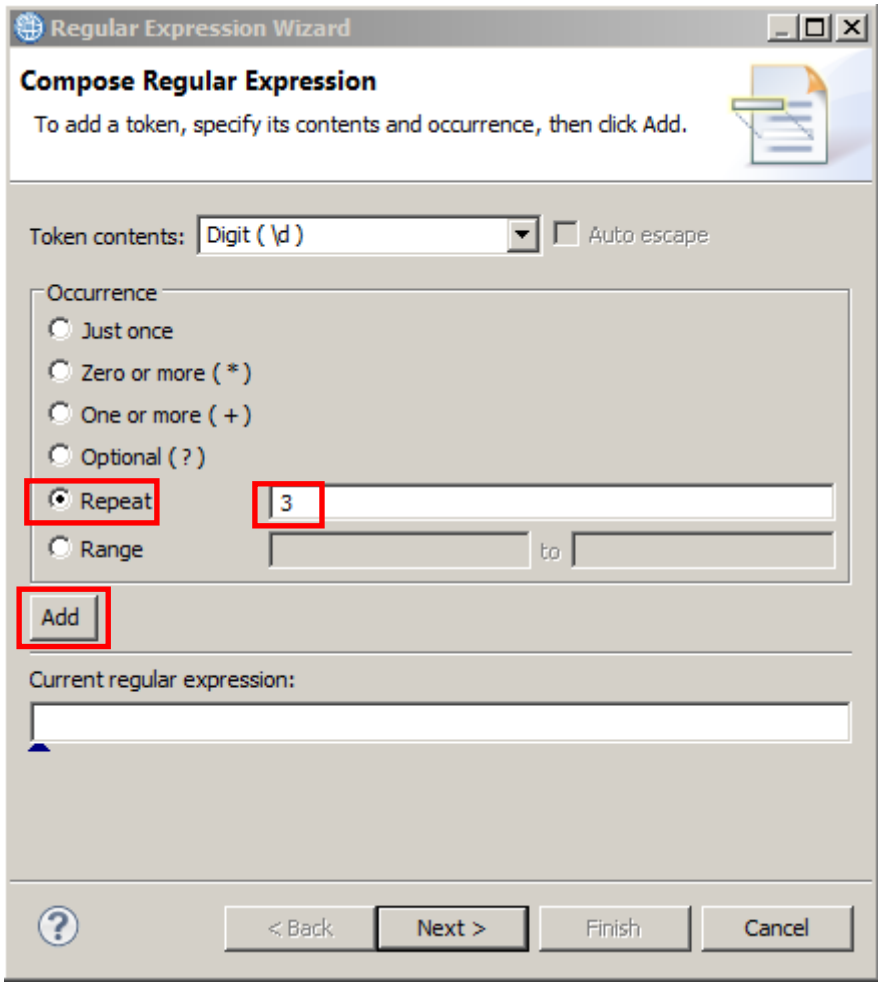
18. Click the "Add.." button to create a regular expression that will define a telephone number pattern.



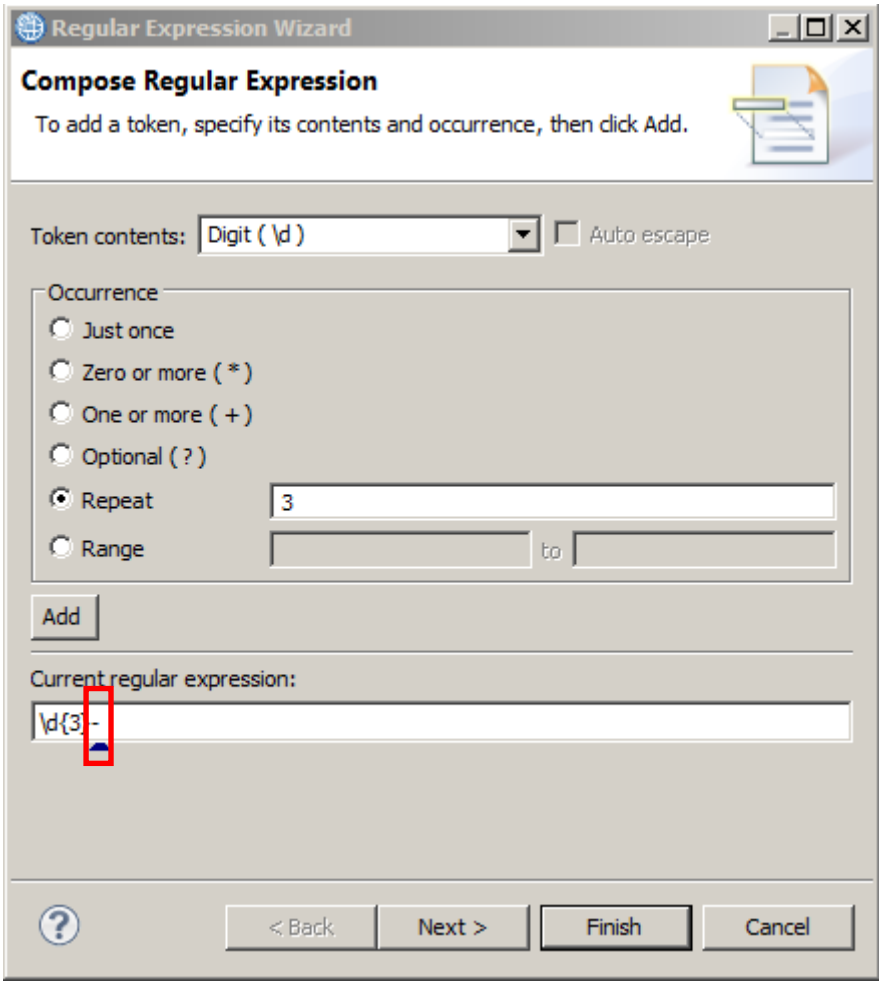
19. In the Regular Expression wizard, select "Digit" from the "Token contents" dropdown.



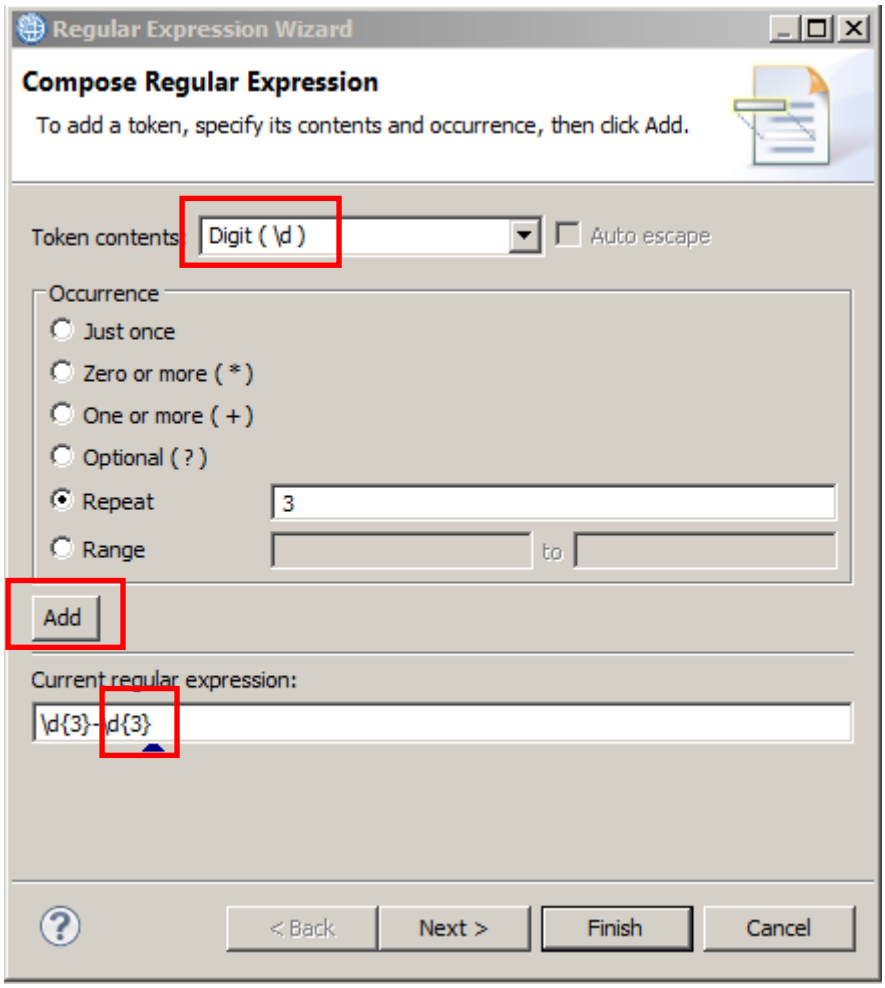
20. Then select the "Repeat" option, enter "3" as its value and click "Add".



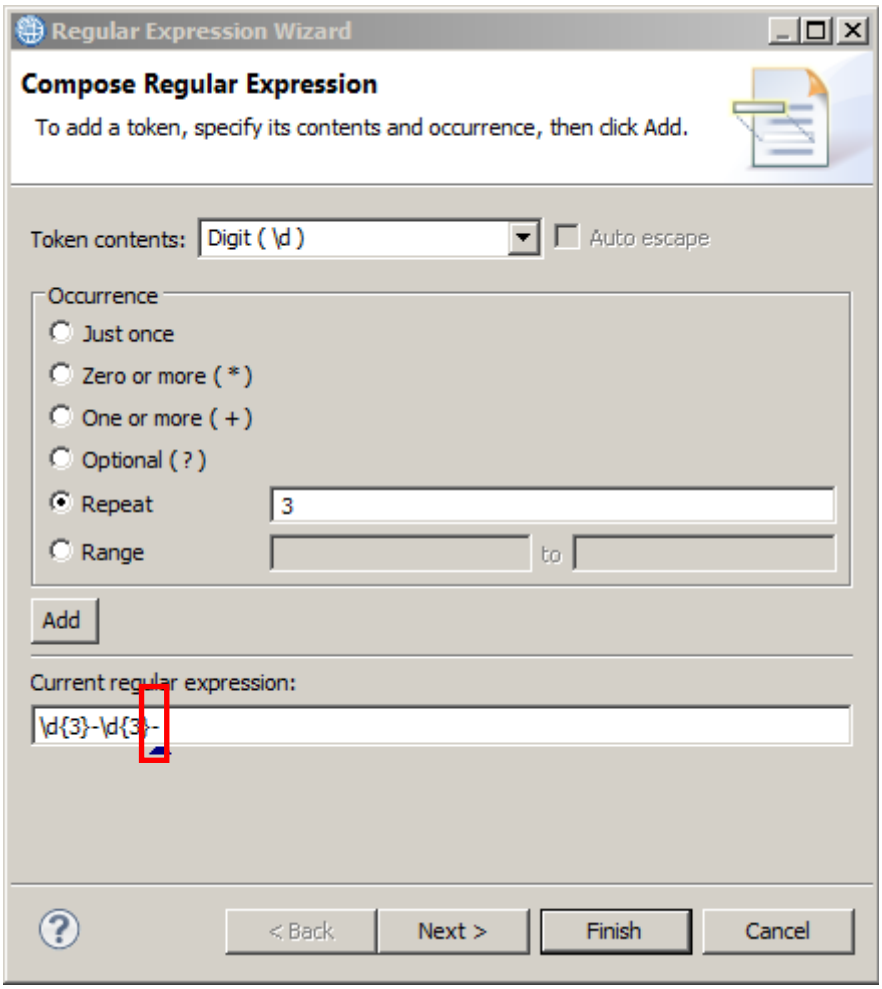
21. In the "Current regular expression" field, enter a hyphen ("-") after the text:



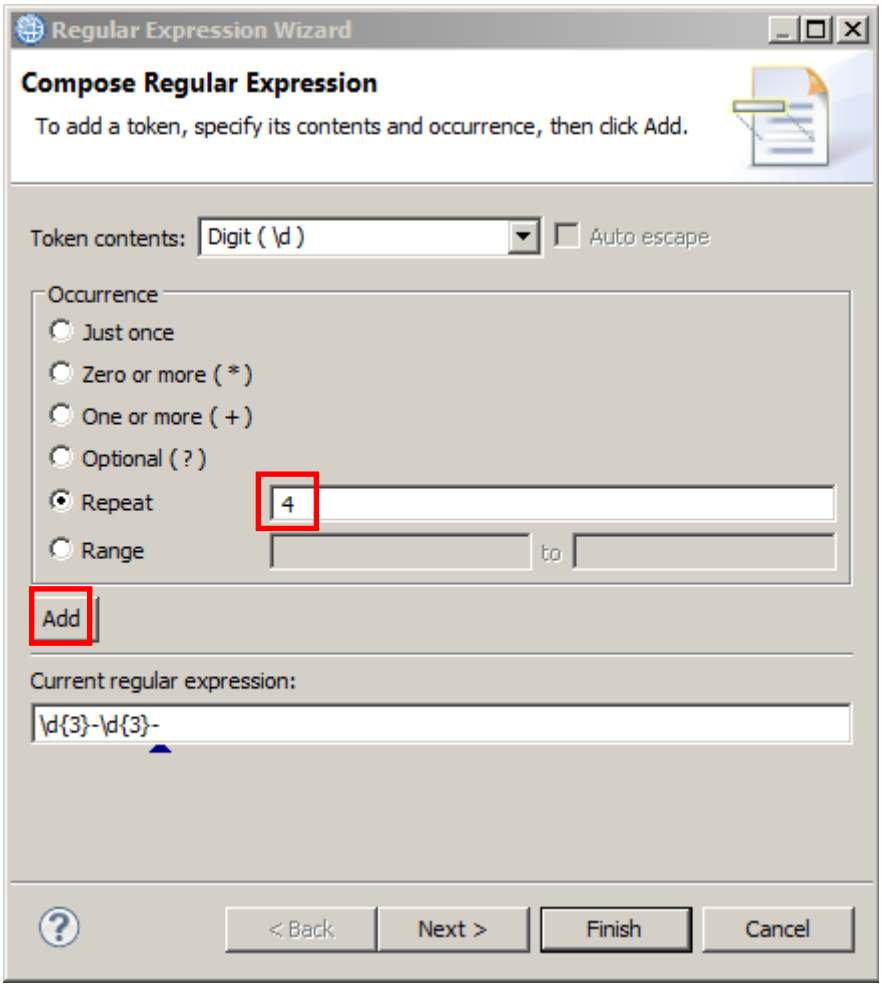
22. Make sure the "token contents" dropdown is set to "Digit" and click the "Add" button again, to add another 3 digits expression.



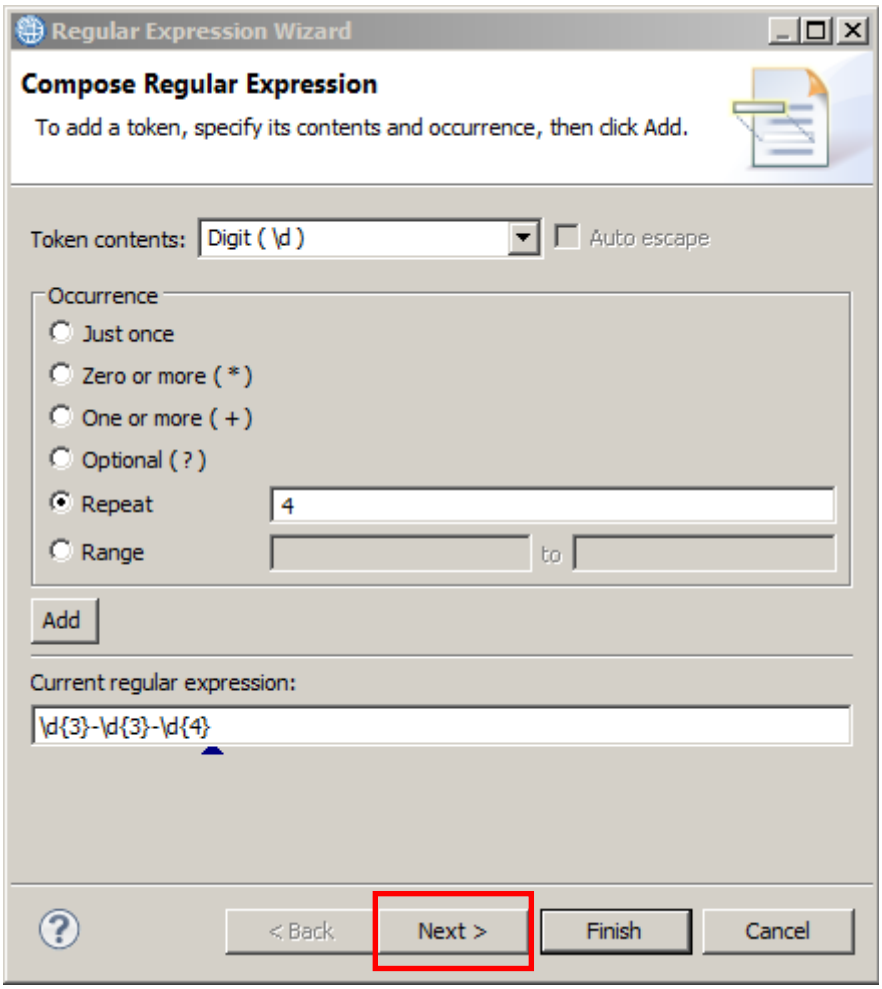
23. In the "Current regular expression" field, enter another hyphen ("-") after the text:



24. Make sure the "token contents" dropdown is set to "Digit", modify the "Repeat" field from "3" to "4" and click the "Add" button again, to add a 4 digits expression.

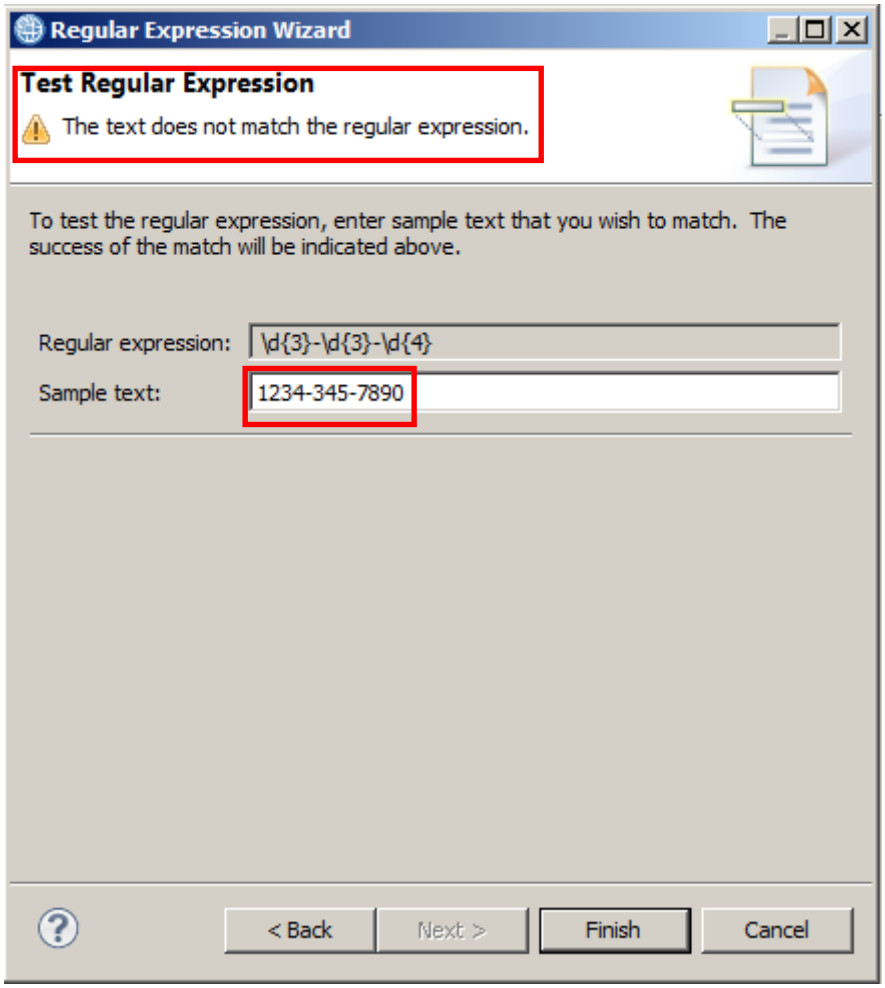


25. Click on the Next button.



26. The Regular Expression Wizard has a testing feature that lets you validate the regular expression you've just built.

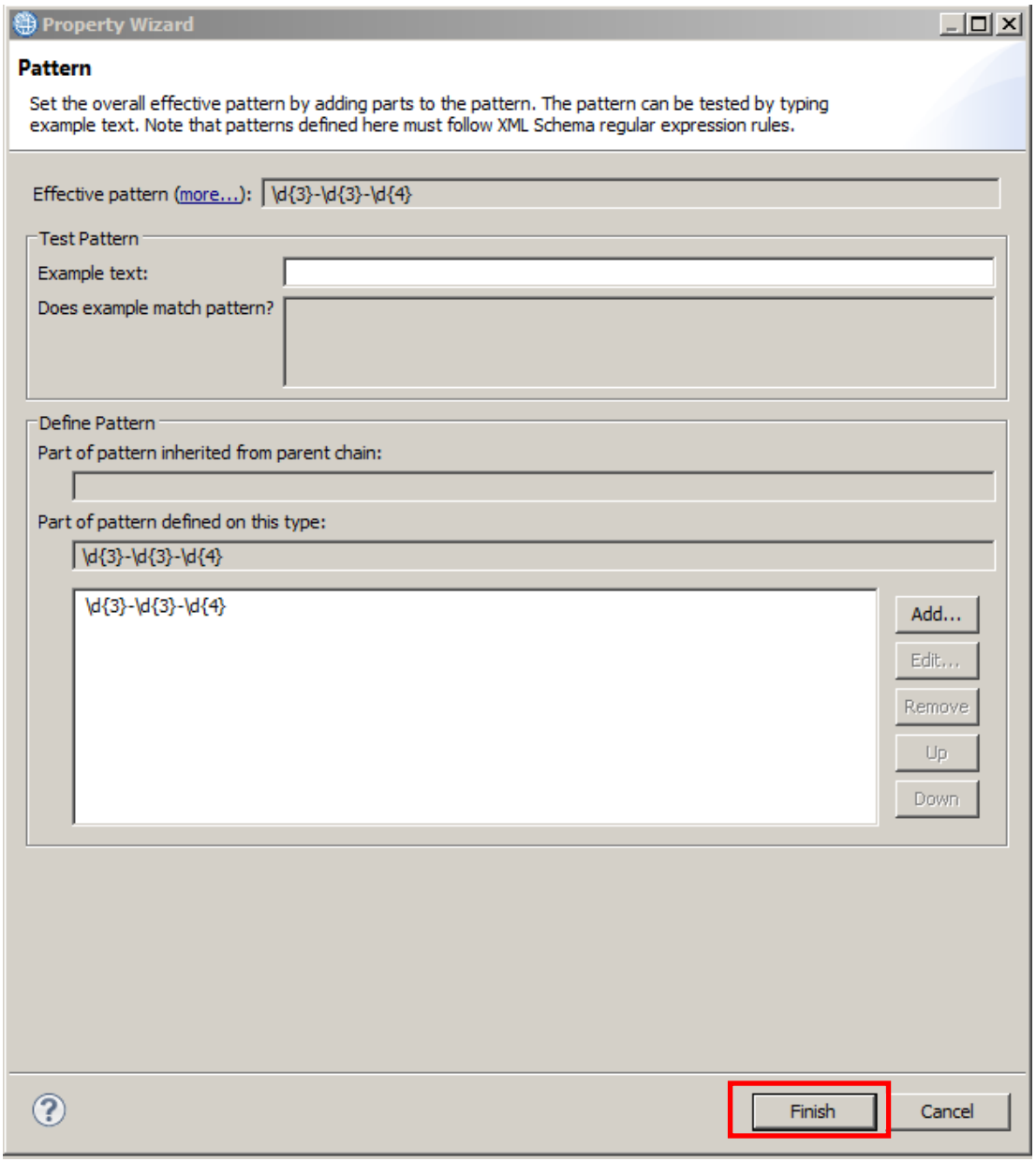
    Enter different strings to test the regular expression, and check that the only valid format is: "3 digits - 3 digits - 4 digits".



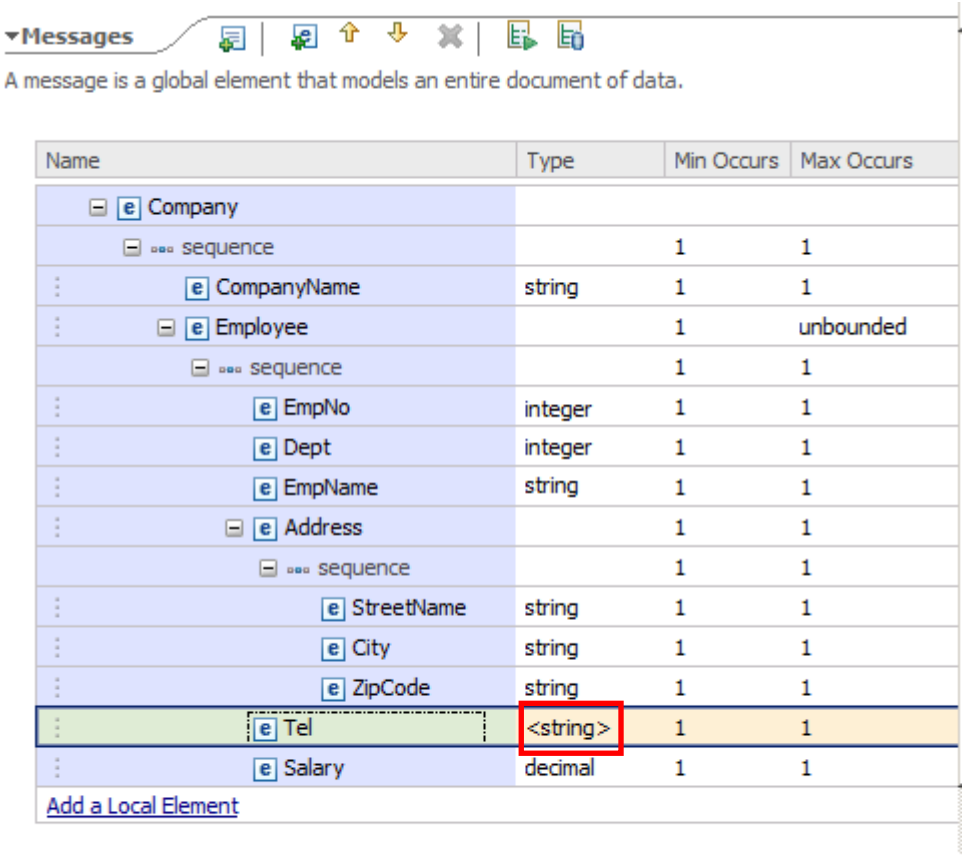
    Then click the Finish button.

27. The Property Wizard is a powerful tool that allows you to build complex regular expressions. In this case you've added just one, but you could create a more complex one by adding several expressions.

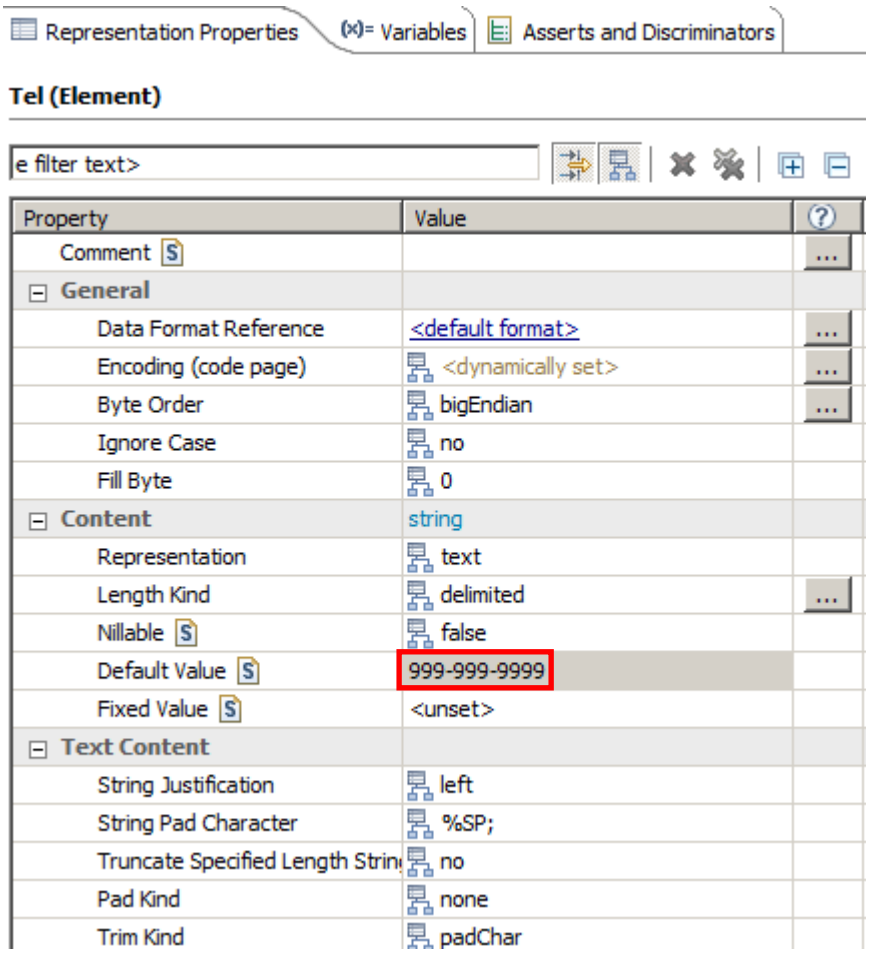
    Click the Finish button.



28. Notice that the "Tel" element's type has changed from "string" to "<string>", an anonymous local restriction of xs:string, in order to carry the pattern facet.



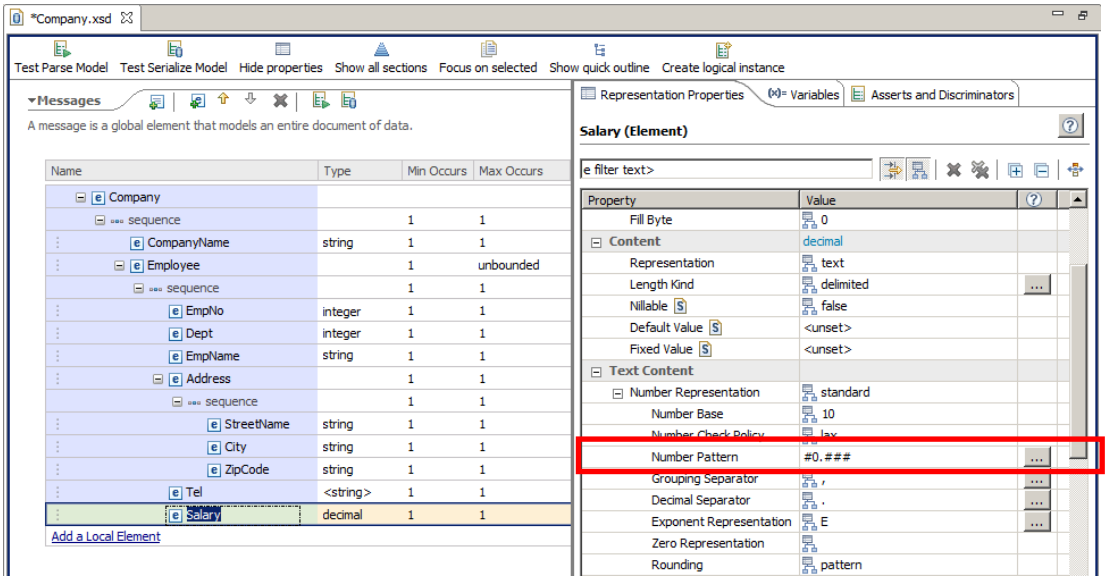
29. Change the "Default Value" of the "Tel" element to a pattern complying value by double-clicking on the "Default Value" row in the Representation Properties (for example: 999-999-9999).



30. Now highlight the "Salary" element and look for the "Text Content" section in the Representation Properties of the DFDL Editor.

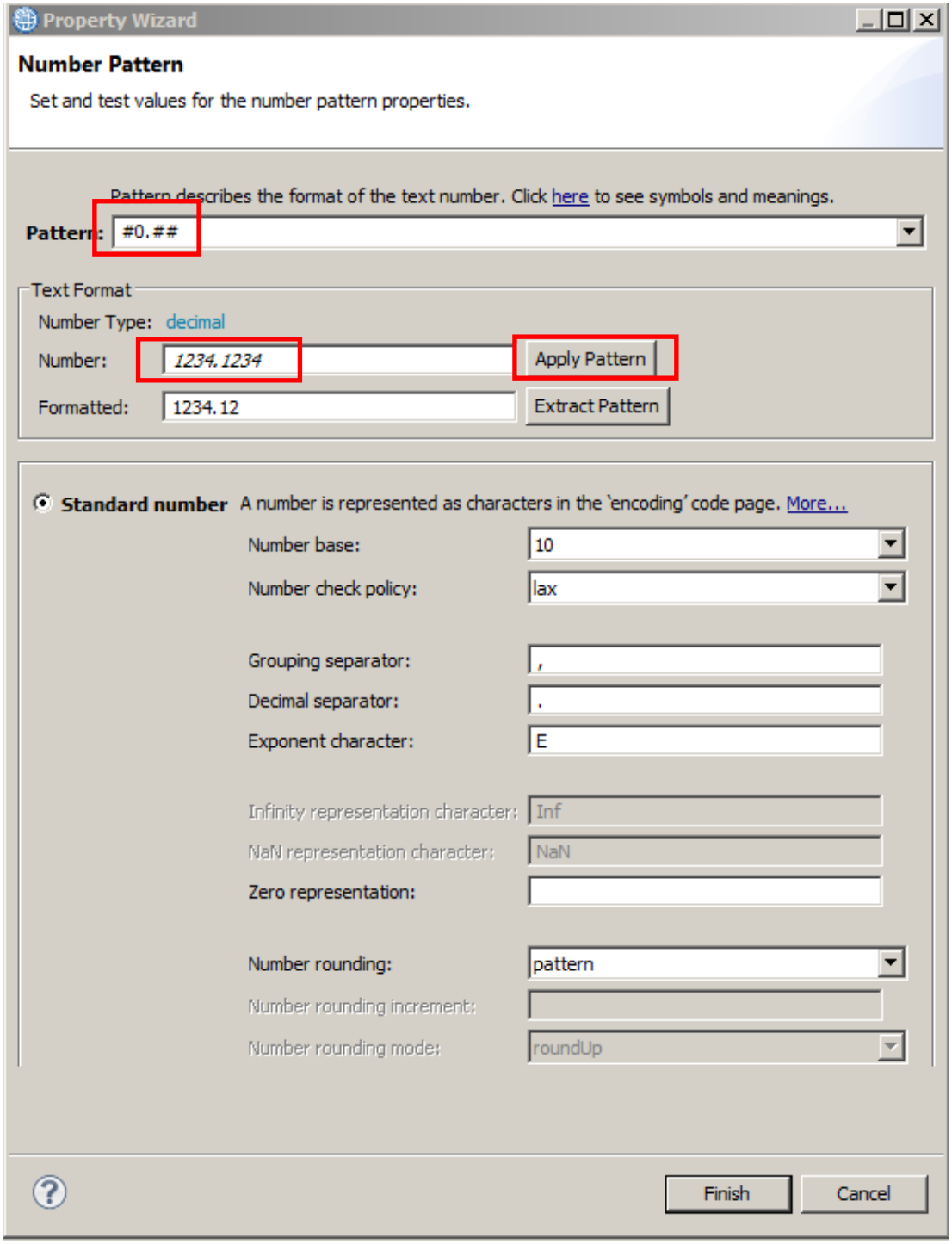
    Expand "Number Representation".

    Click on the button (three dots) next to "Number Pattern".



31. In the Number Pattern Property Wizard, change the Pattern to "#0.##" (delete the final #).

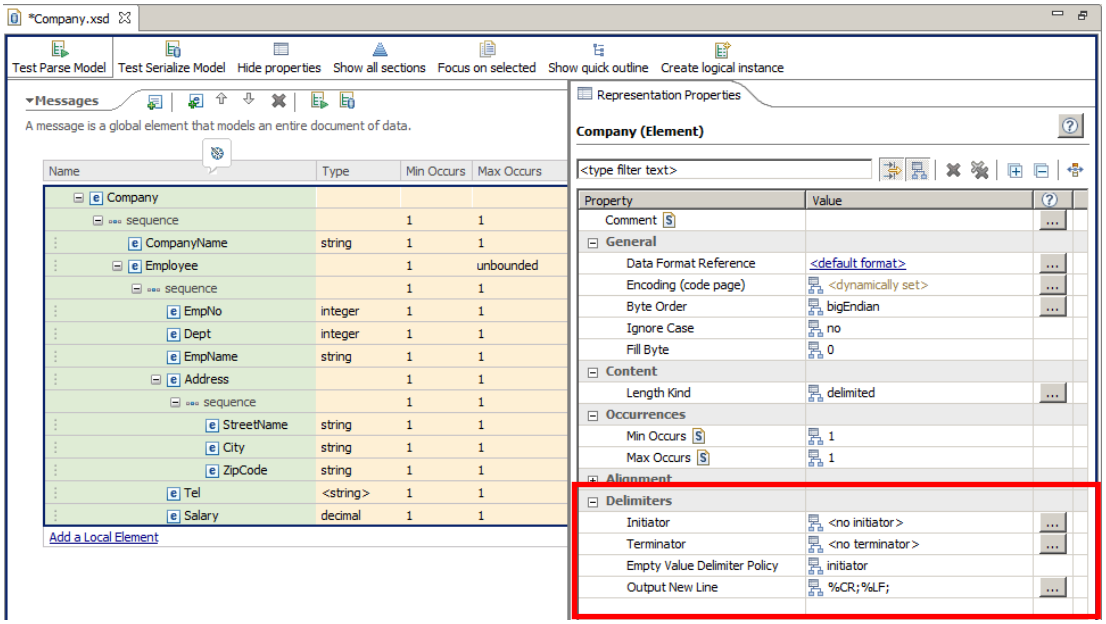
    Enter "1234.1234" in the "Number" field in the Text Format section. Click on the "Apply Pattern" button to test the Number Pattern. Notice that the number was changed from "1234.1234" to "1234.12" to comply with the defined number pattern.



Click Finish.

32. Next you will define the Initiators, Terminators and Separators for the Message Model.

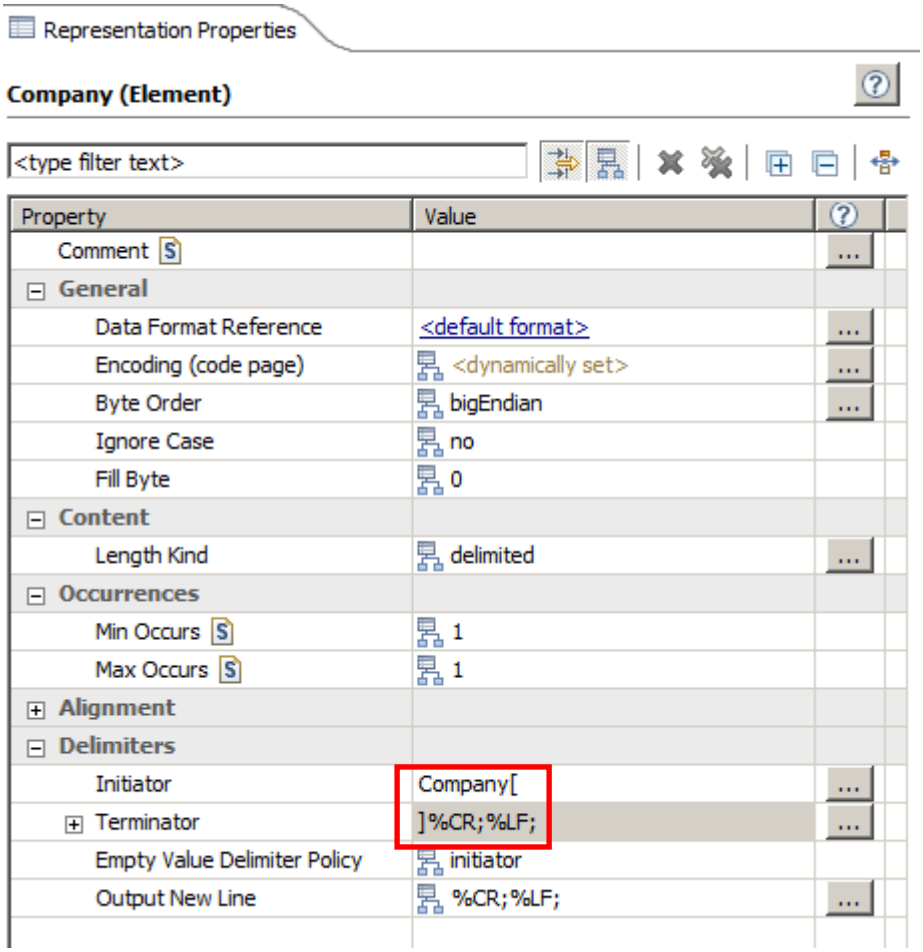
    Click on the "Company" element (message root) and look at the "Delimiters" section in the Representation properties view in the DFDL Editor.



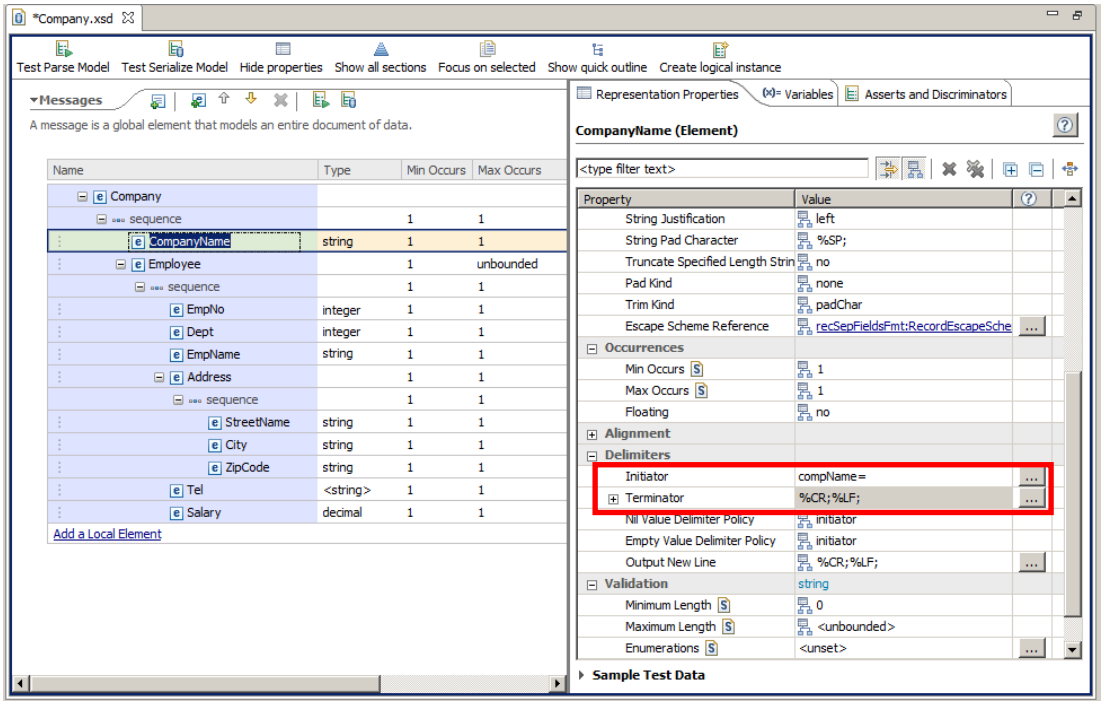
33. Enter "Company[" as the Initiator, and "]%CR;%LF;" as the terminator. (Do not include the quotation marks).

    Hint: after you have entered the “]”, you can use Ctrl-Space to use the Toolkit Content Assist editor, and select the CR and LF values.

    This definition implies that the record starts with a "Company[" tag and ends with a "]%CR;%LF;" tag.

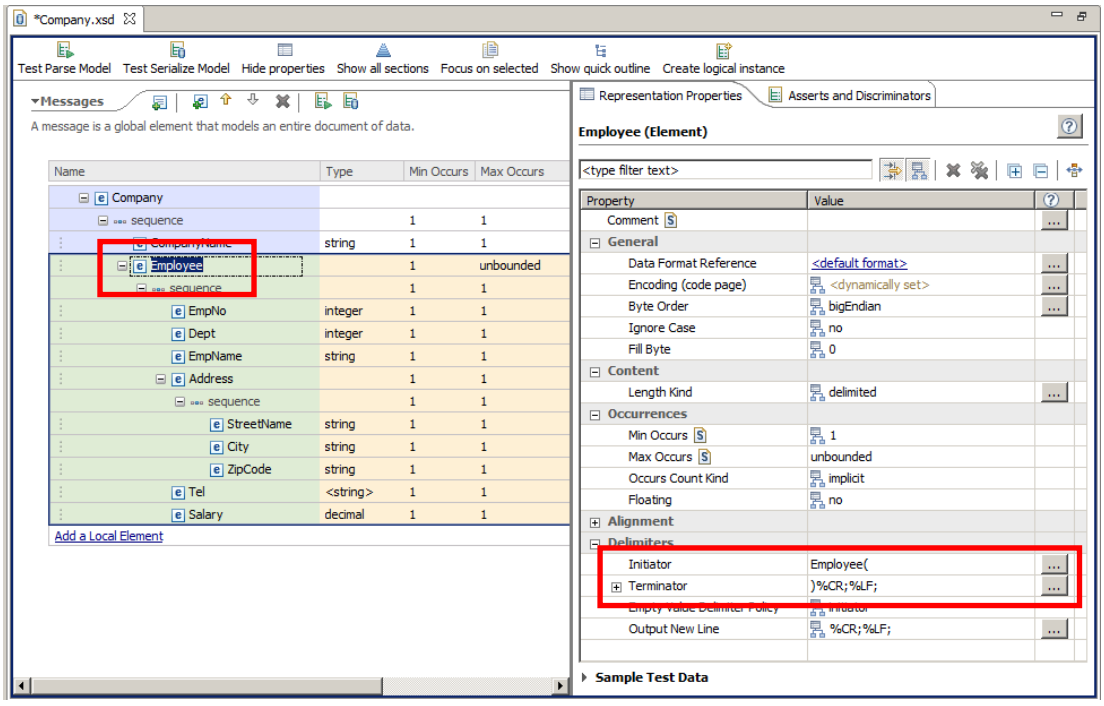


34. Click on the "CompanyName" element, and in the "Delimiter" section of the Representation properties view , enter "compName=" as the Initiator and "%CR;%LF;" as the Terminator:

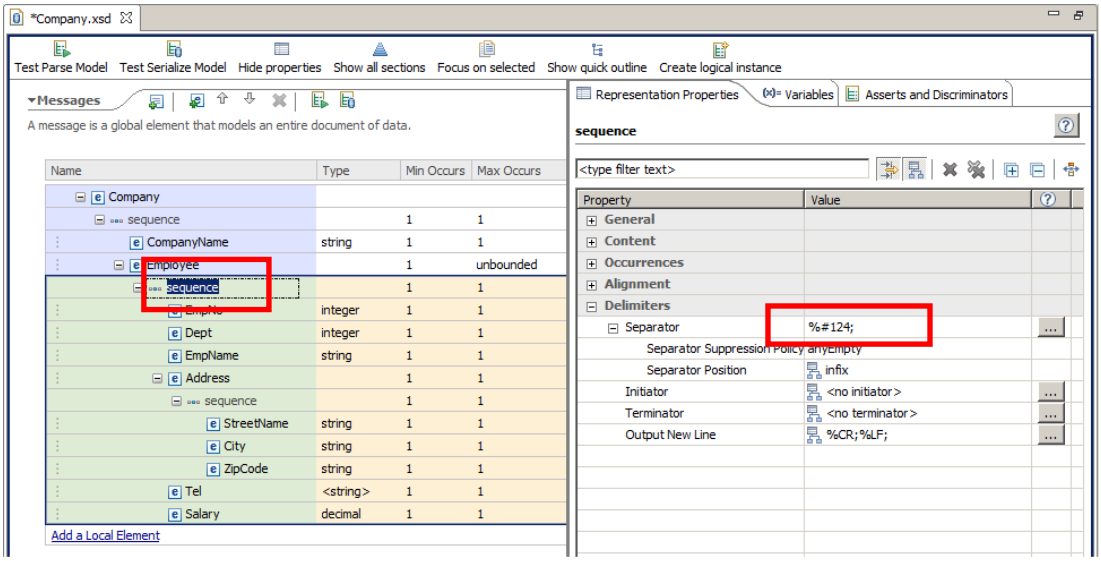


35. Click on the "Employee" element, and in the "Delimiter" section of the Representation properties view, set the Terminator value to ")%CR;%LF;". Make sure you don't miss the "**)**" at the start of the terminator string.

    Make sure the initiator is set to "Employee(" , it should have been completed automatically by the wizard at the beginning.



36. Now click on the <sequence> content of the Employee element and in the Representation Properties view, check that the Separator is set to "%#124;" (the wizard should have completed it).

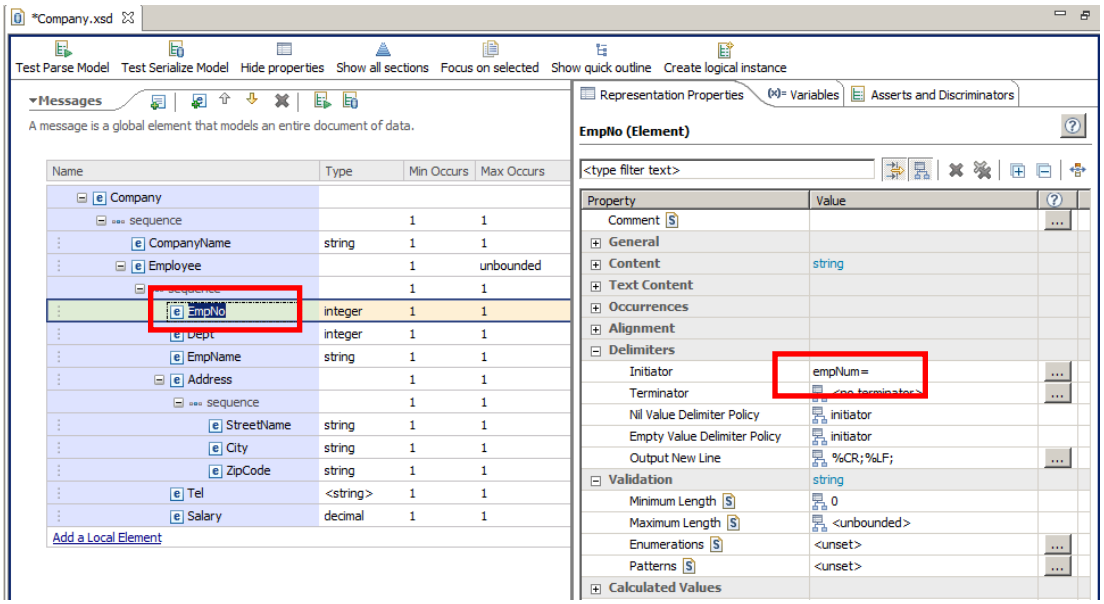


This Separator defines that all the fields inside the "Employee" structure are separated by the "|" character.

37. For the fields in the Employee structure, change the Initiator of each one to the following:

Element	Initiator
EmpNo	empNum=
Dept	dept=
EmpName	empName=
Address	Addr:
Tel	tel=
Salary	sal=

Note the Address initiator uses a colon, not an “equals”.



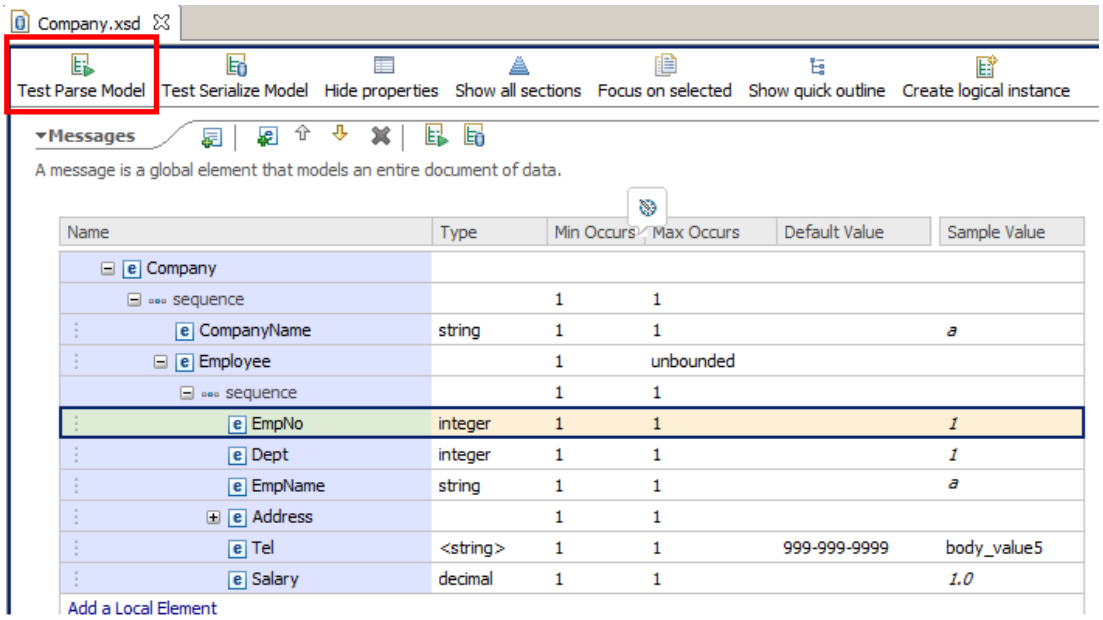
38. Save your DFDL Schema by pressing Ctrl+S or File->Save. When saved, the DFDL Schema is validated and if any errors (or warnings) are found, they will appear in the Problems view.

    Make sure there are no errors in the Problems view.

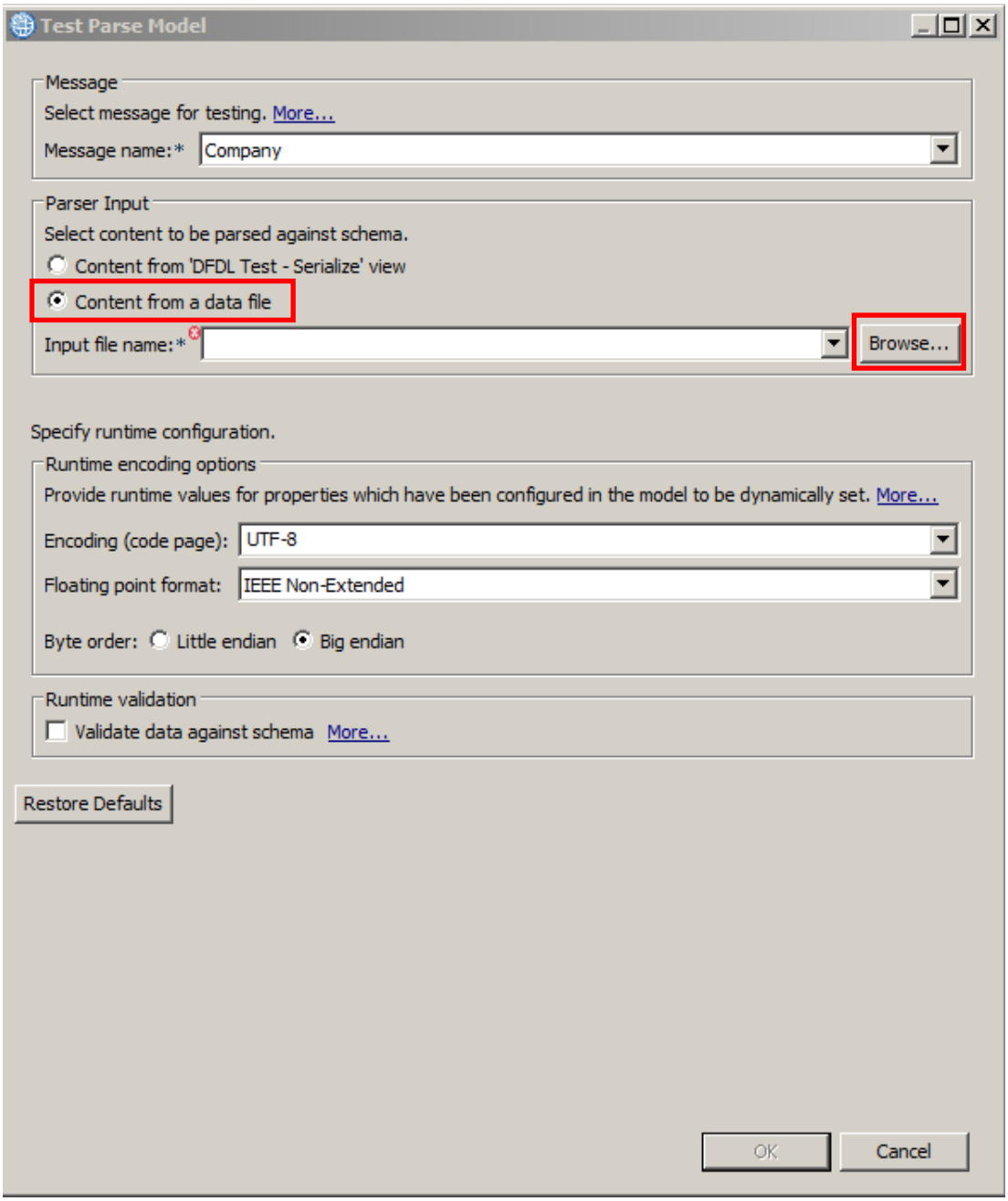
4. Testing the Message Model

Now that the message model is complete, you can test parse it against a sample data file.

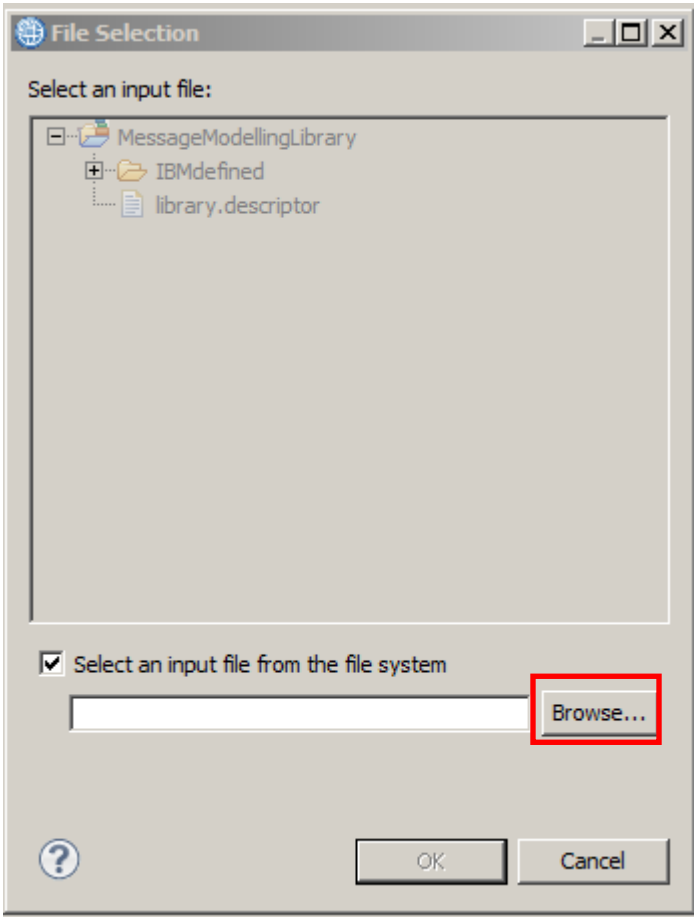
Click on the "Test Parse Model".



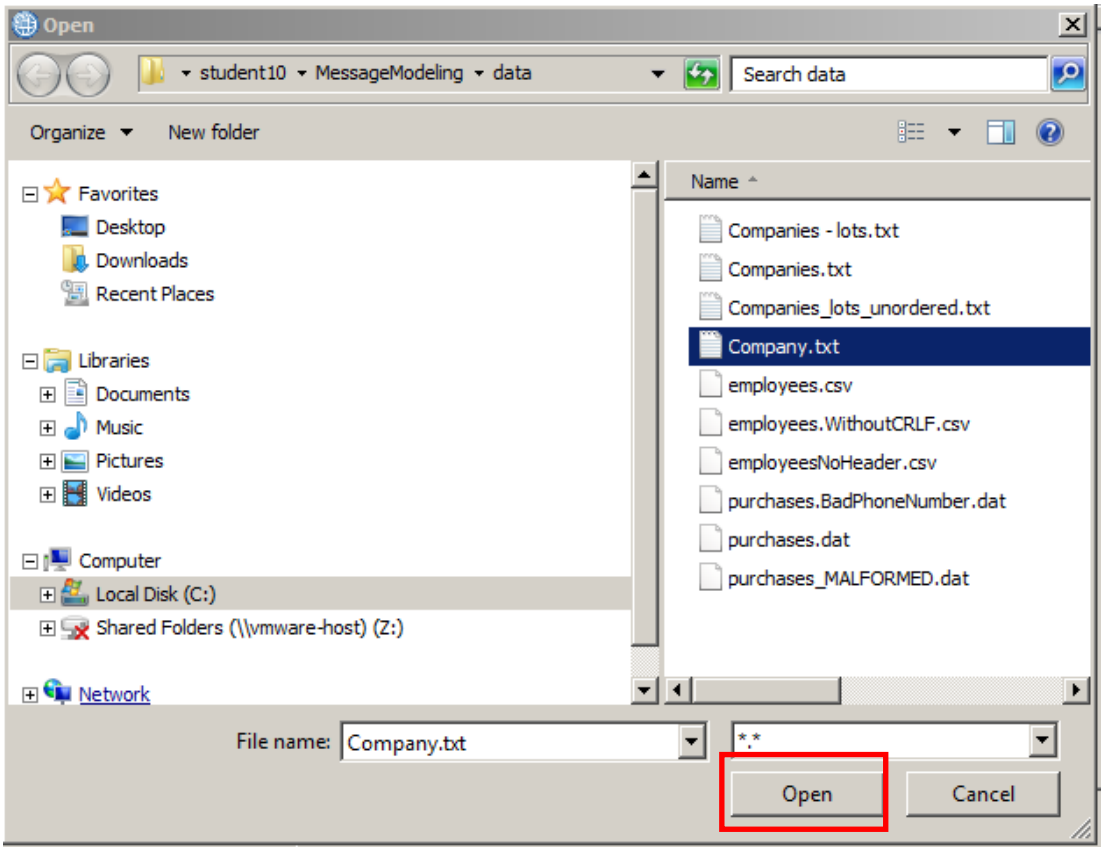
2. Select the "Content from a data file" option. Click the Browse button.



3. Check the "Select an input from the file system" checkbox, and click the Browse button.

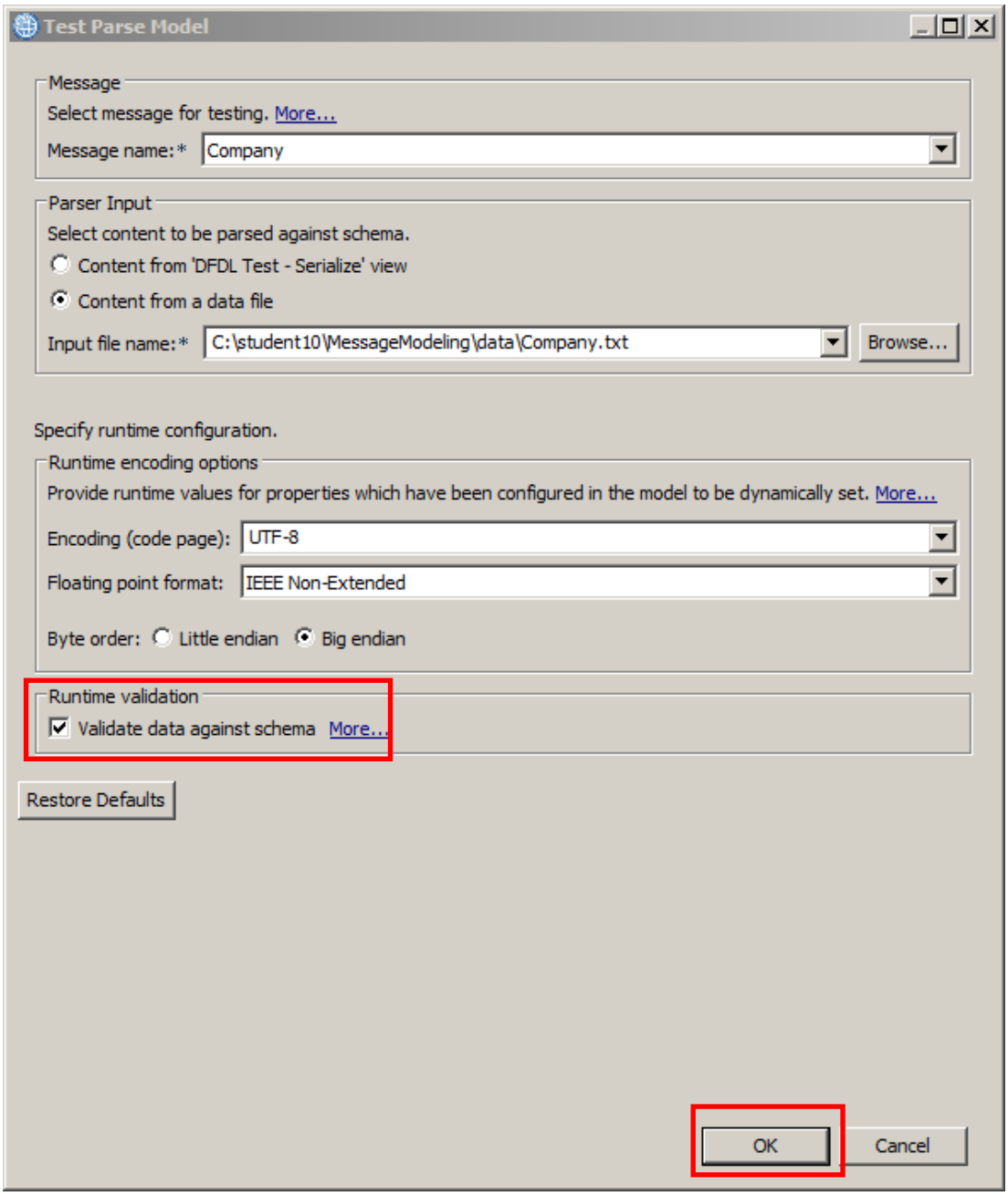


4. Browse to the "C:\student10\MessageModeling\data" directory and select "Company.txt". Click Open, and then OK.

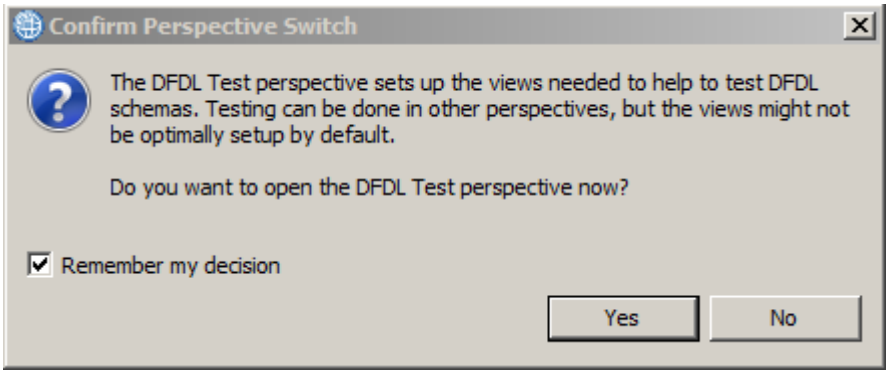


5. Check "Validate against schema" to enable validation (to test the telephone pattern you defined for validation of the "Tel" field).

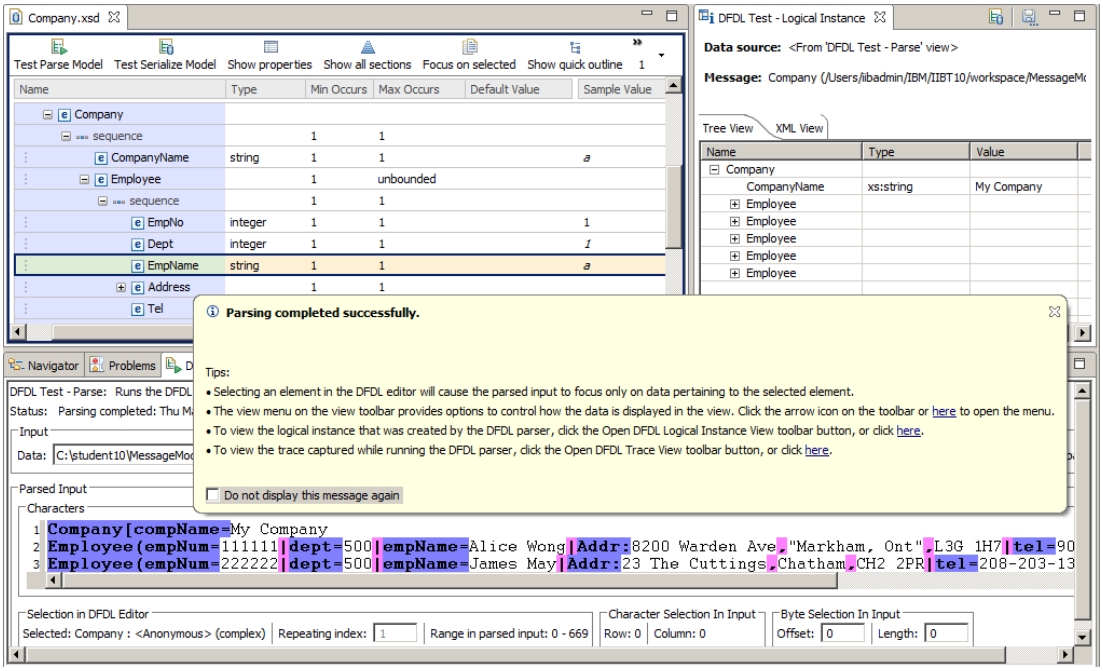
   Click OK.



6. If the "Confirm Perspective Switch" window appears, check the "Remember my decision" checkbox and click Yes.

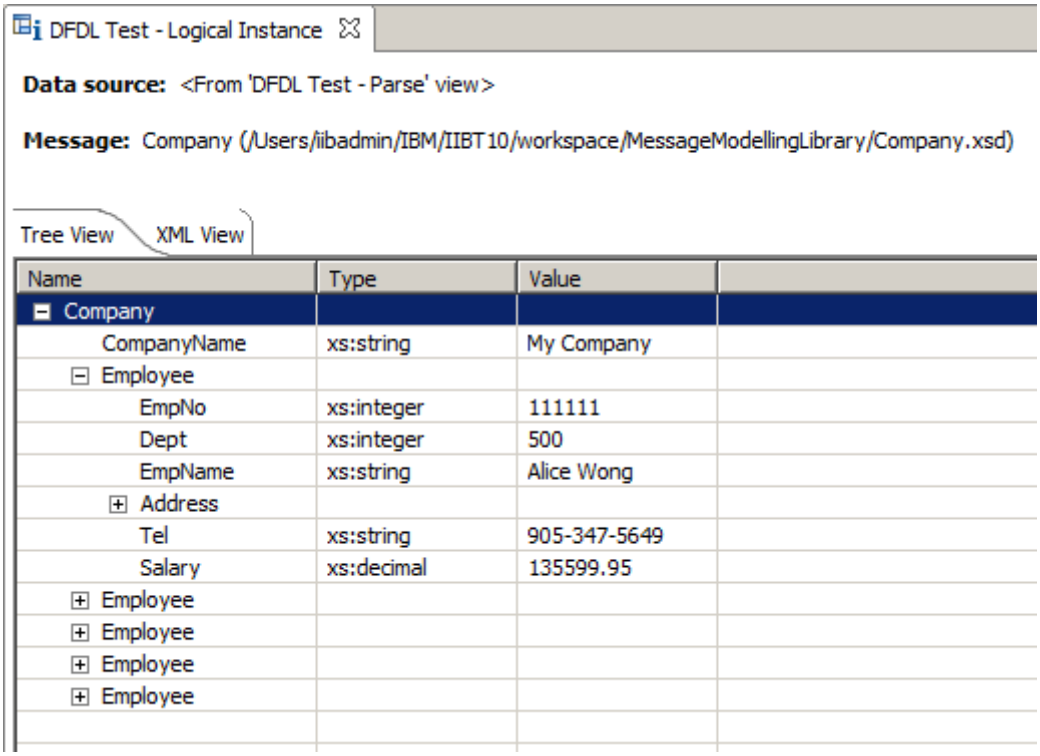


7. In the "DFDL Test" perspective, "DFDL Test - Parse" view, a message bubble appears indicating the parsing was successful.



Close the message by clicking on the "X".

8. Go to the "DFDL Test - Logical Instance" view, and take a look at the parsed message tree and check if it is correct.



Name	Type	Value
Company		
CompanyName	xs:string	My Company
Employee		
EmpNo	xs:integer	111111
Dept	xs:integer	500
EmpName	xs:string	Alice Wong
Address		
Tel	xs:string	905-347-5649
Salary	xs:decimal	135599.95
Employee		
Employee		
Employee		
Employee		

END OF LAB GUIDE