

# IBM Integration Bus

## Message Modeling with DFDL

### Lab 3

Record-oriented, tagged, delimited text

June, 2013

Hands-on lab built at product  
code level version 9.0.0.0

**1. INTRODUCTION ..... 4**  
    1.1 LAB PREPARATION ..... 4  
    1.2 LAB SCENARIO ..... 4  
**2. BUILD THE MESSAGE MODEL ..... 5**  
**3. REFINING THE MESSAGE MODEL ..... 12**  
**4. TESTING THE MESSAGE MODEL ..... 38**

# 1. Introduction

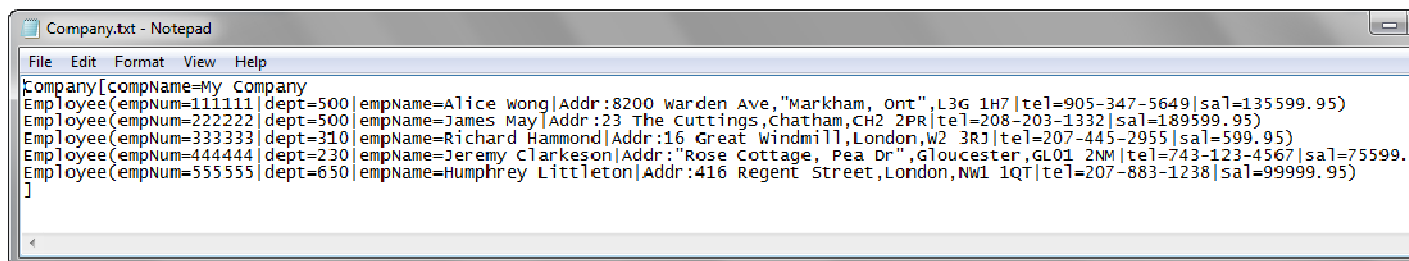
## 1.1 Lab preparation

To run this lab, unzip the supplied file MessageModelling.zip into the directory c:\student 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.

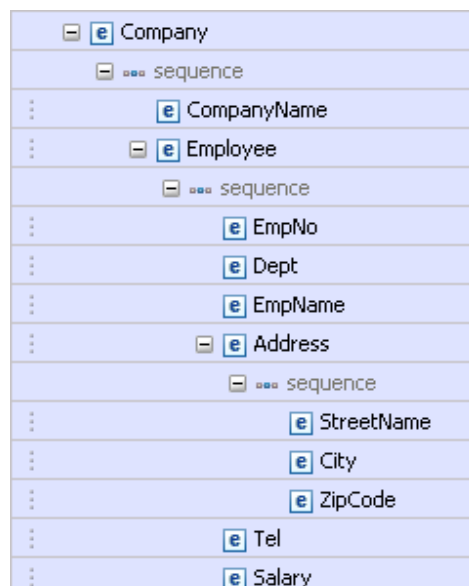
In this lab you will build a message model capable of parsing this tagged / delimited file:



```

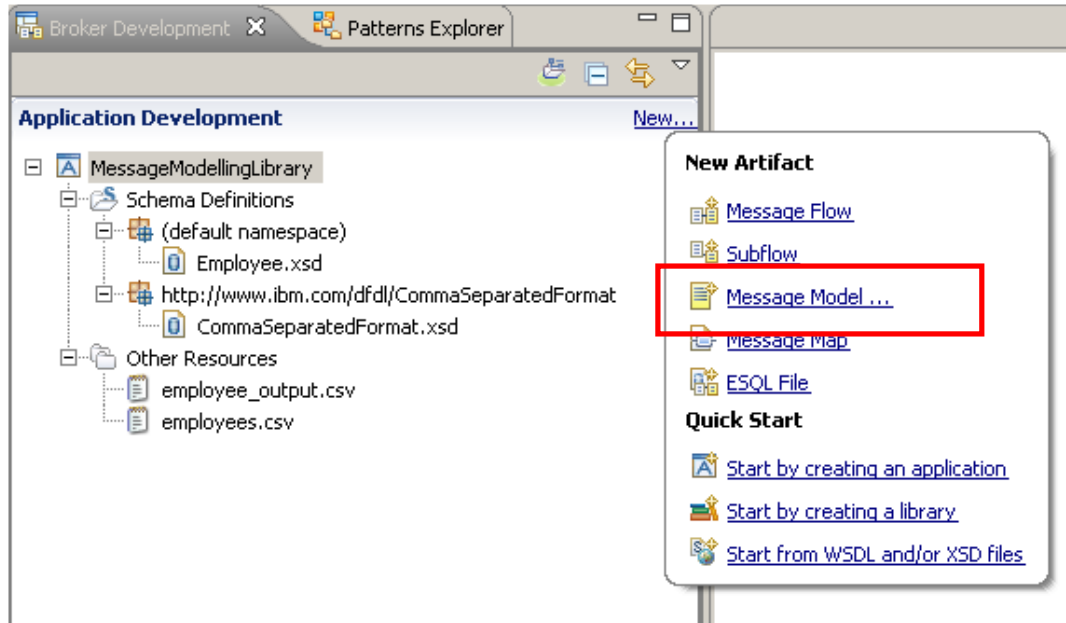
Company[compName=My Company
Employee(empNum=111111|dept=500|empName=Alice Wong|Addr:8200 warden Ave,"Markham, ont",L3G 1H7|tel=905-347-5649|sal=135599.95)
Employee(empNum=222222|dept=500|empName=James May|Addr:23 The Cuttings,Chatham,CH2 2PR|tel=208-203-1332|sal=189599.95)
Employee(empNum=333333|dept=310|empName=Richard Hammond|Addr:16 Great Windmill,London,W2 3RJ|tel=207-445-2955|sal=599.95)
Employee(empNum=444444|dept=230|empName=Jeremy Clarkson|Addr:"Rose Cottage, Pea Dr",Gloucester,GL01 2NM|tel=743-123-4567|sal=75599.95)
Employee(empNum=555555|dept=650|empName=Humphrey Littleton|Addr:416 Regent Street,London,Nw1 1QT|tel=207-883-1238|sal=99999.95)
]
  
```

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

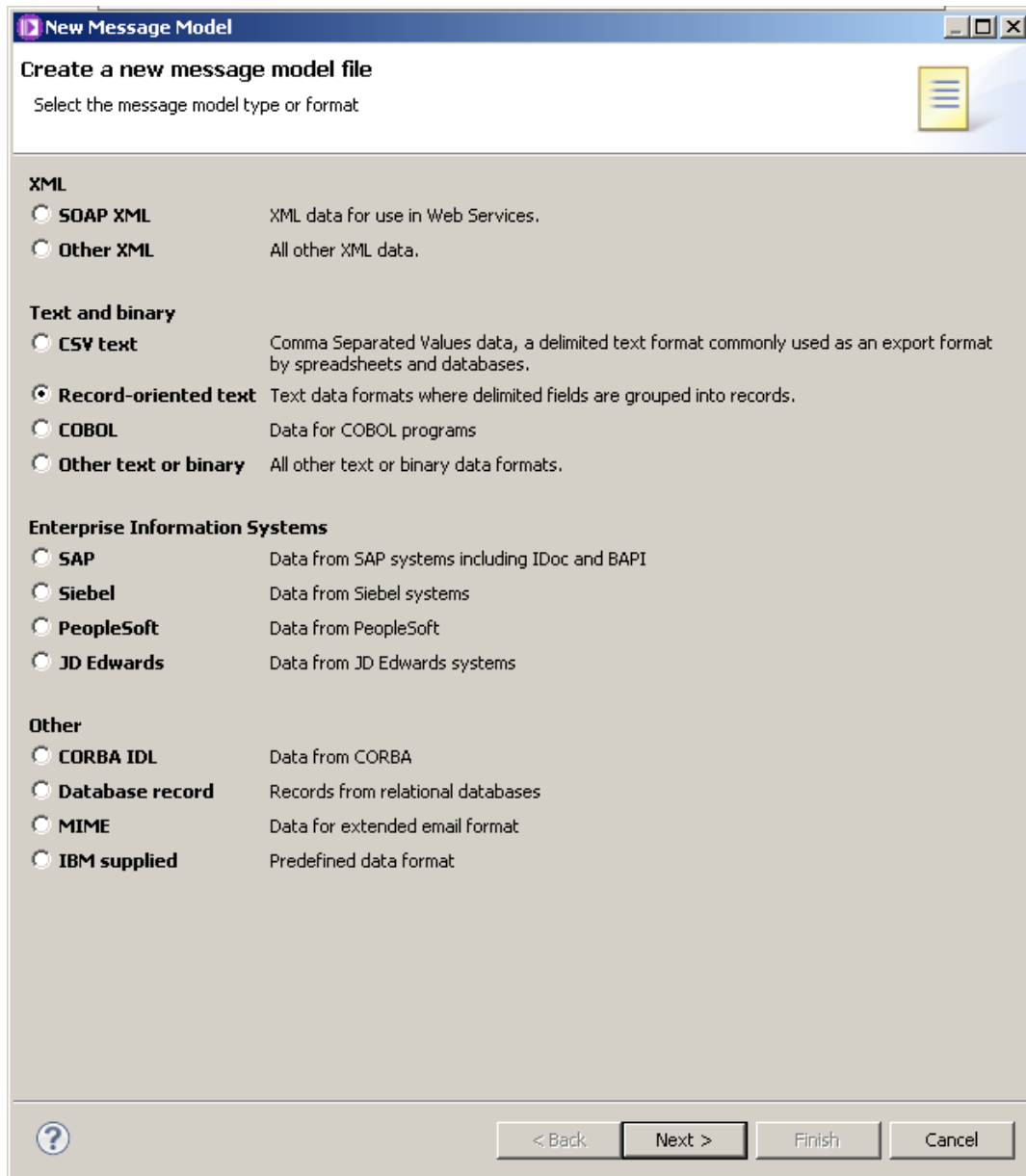


## 2. Build the Message Model

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

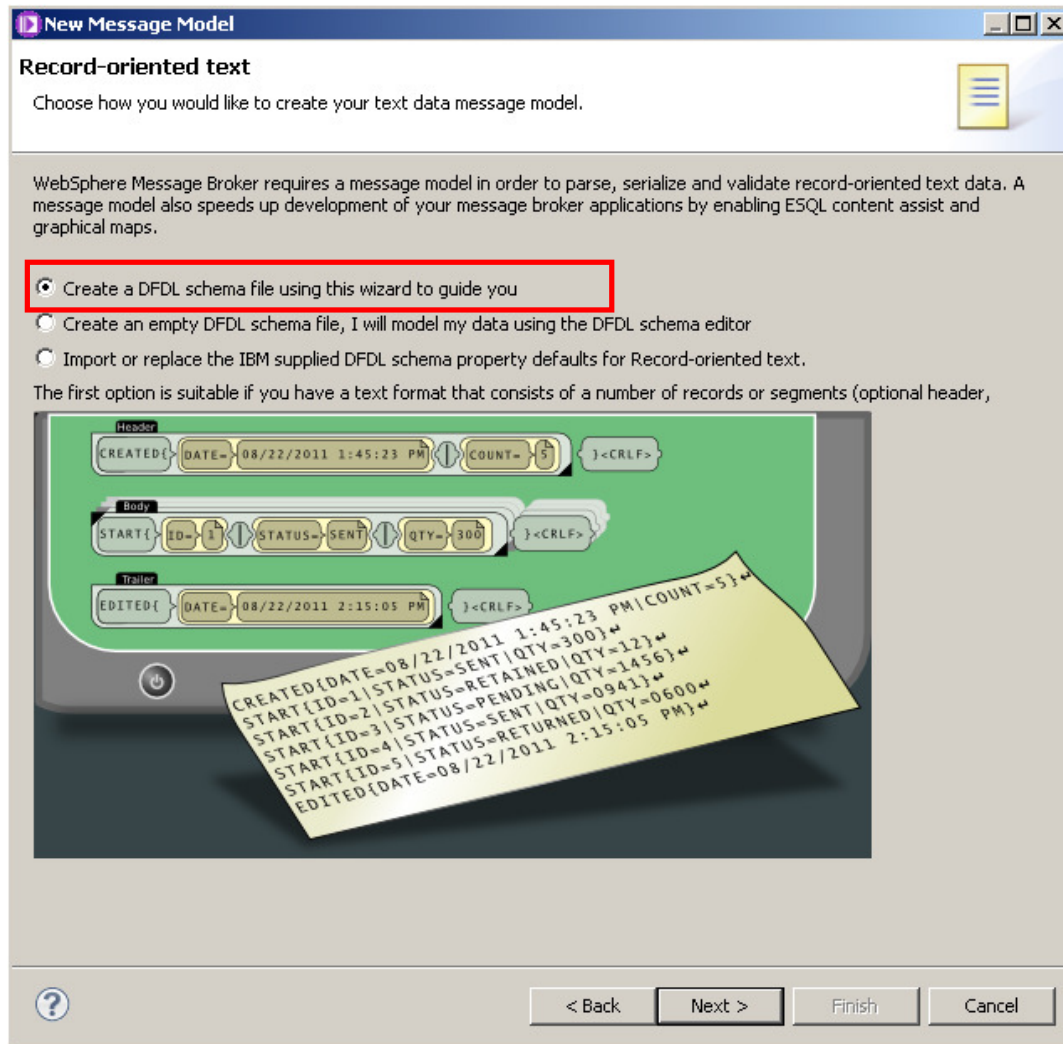


2. In the "New Message Model" window, select "Record-oriented text" and click Next.



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.

**New Message Model**

**Create a Data Format Description Language (DFDL) Schema**  
Specify the location and name of the DFDL schema, and specify the name of the message.

Application or Library:  Browse... New...

Folder:  Browse... New...

DFDL schema file name:

Message name:

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.

**New Message Model**  
Configure schema for data formatted as records and fields  
Provide setting for new DFDL schema that represent record-oriented data.

Record settings  
End of record character: Carriage Return & Line Feed - %CR;%LF;  
(Blank records will be skipped)  
 The first record is a header  
 The last record is a trailer

Header fields | **Body fields** | Trailer fields  
Record initiator: Employee(  
Number of fields: 6

Field settings  
 Separated by: | - %#124; (UTF-8: 0x7C) (UTF-16: 0x007C)  
 Fixed length  
 All fields have an initiator  
 Create default values for fields

Encoding code page options:  
 Dynamic (provided to the processor by the application at runtime)  
 Fixed UTF-8

Global settings  
Escape scheme: Default escape scheme

< Back Next > Finish Cancel

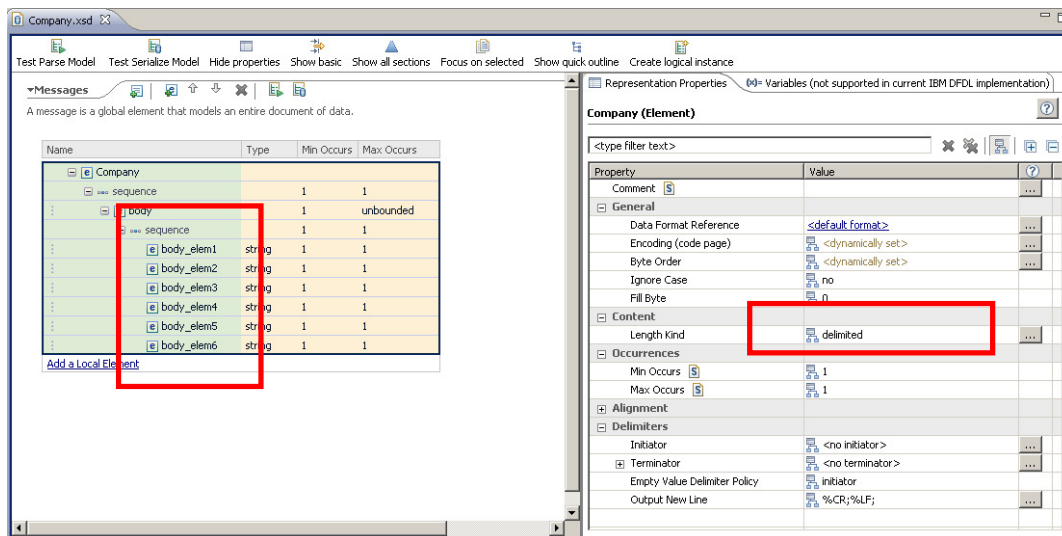
Leave "Separated by" as "|" (pipe) and "All fields have an initiator" checked. These default values match the required for the message model.

Click Finish.

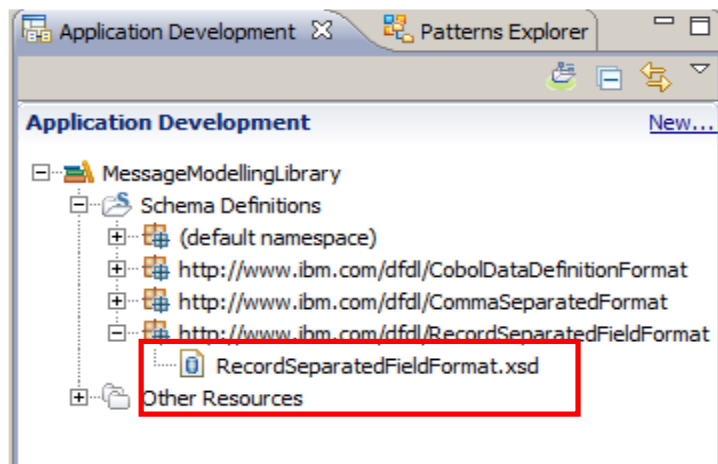


- When the wizard finishes, the DFDL Editor will open with the generated Company.xsd schema file.

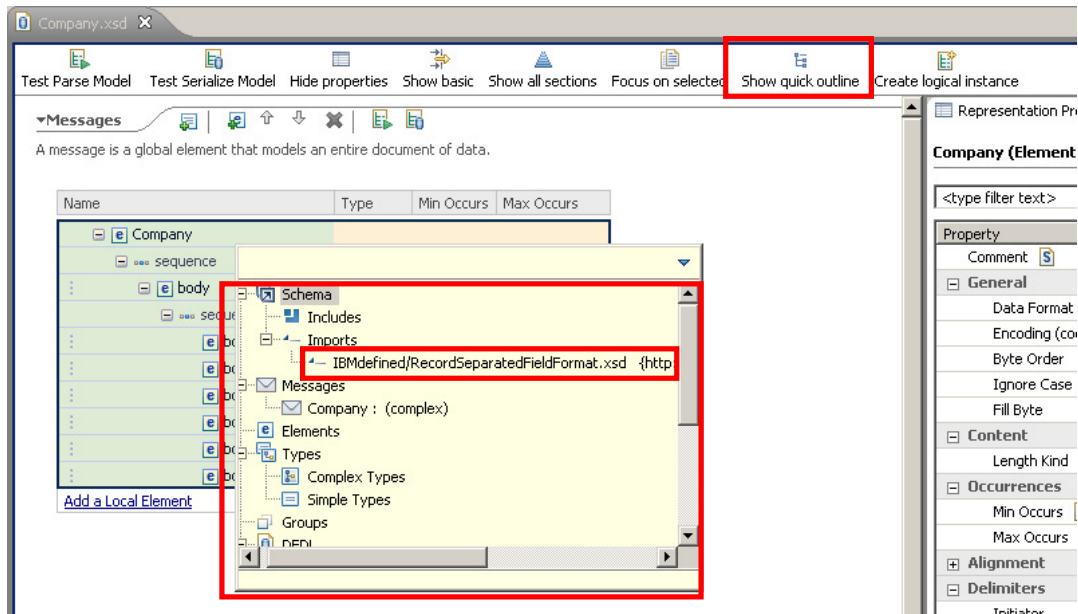
Note that as you defined it in the wizard, the model has six fields, "Length Kind" property is set to delimited, and Separators and Initiators have been set accordingly.



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.



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

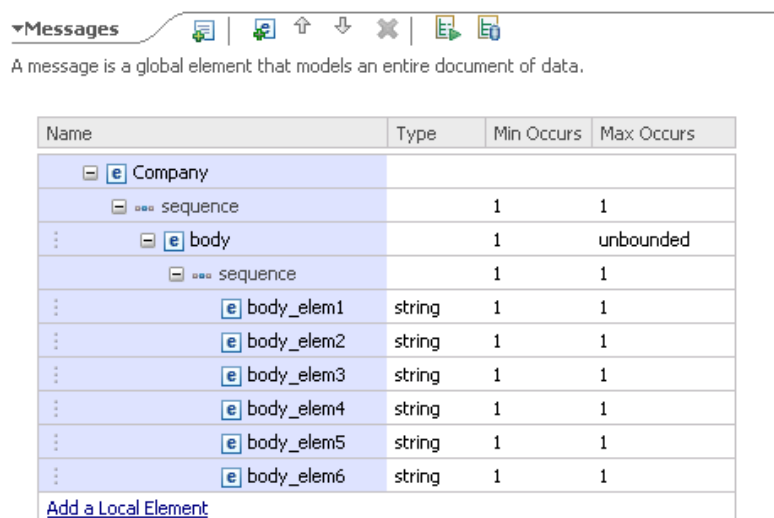


The Outline view will appear with a high level view of the elements of your message model. If you click on any of them, the editor will focus on it.

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

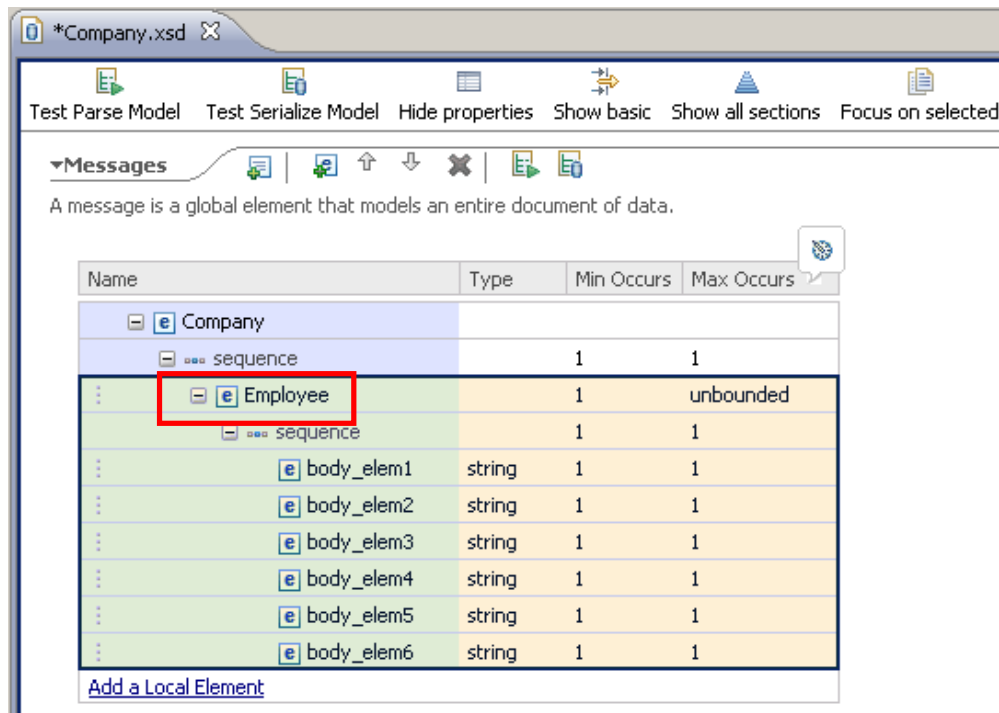
To close the Outline pop-up, click anywhere else on the editor window.

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.

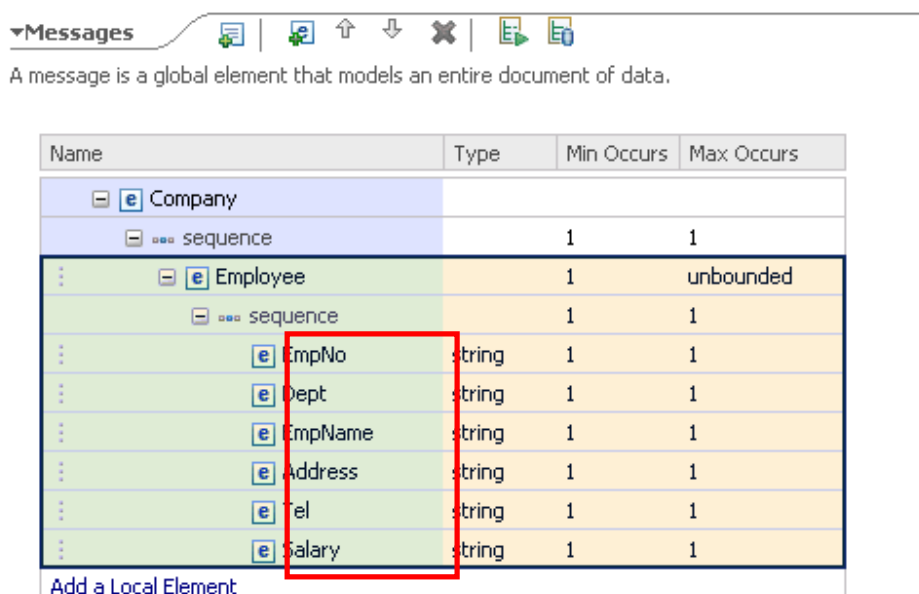


A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
[-] e Employee		1	unbounded
[-] ... sequence		1	1
[-] e body_elem1	string	1	1
[-] e body_elem2	string	1	1
[-] e body_elem3	string	1	1
[-] e body_elem4	string	1	1
[-] e body_elem5	string	1	1
[-] e body_elem6	string	1	1

[Add a Local Element](#)

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.



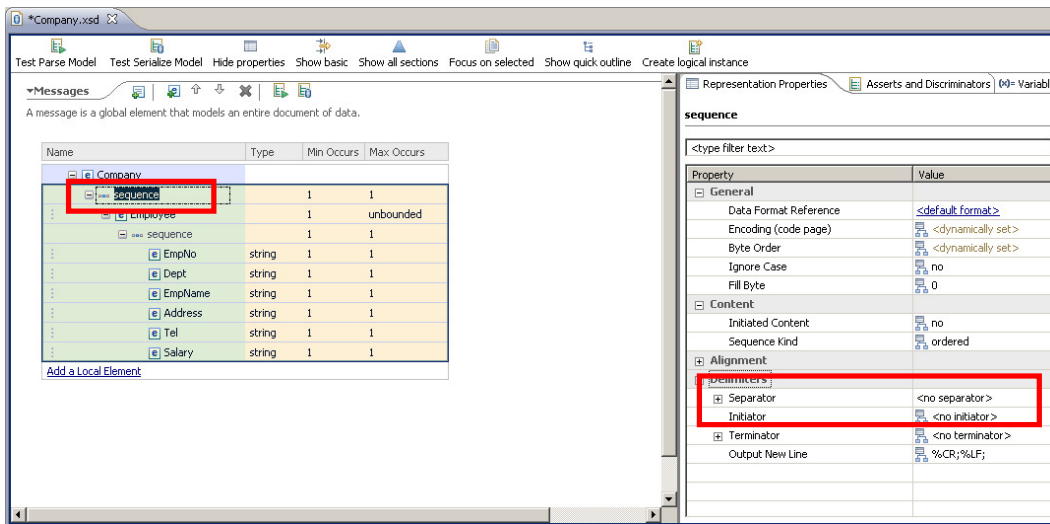
A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
[-] e Employee		1	unbounded
[-] ... sequence		1	1
[-] e EmpNo	string	1	1
[-] e Dept	string	1	1
[-] e EmpName	string	1	1
[-] e Address	string	1	1
[-] e Tel	string	1	1
[-] e Salary	string	1	1

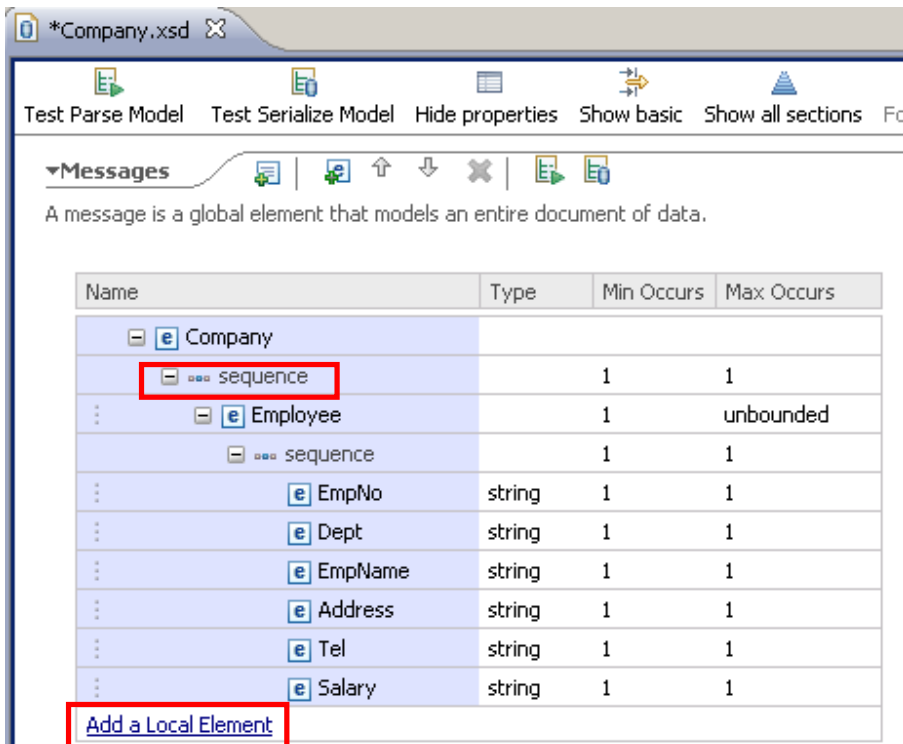
[Add a Local Element](#)

- 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).



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



5. Name the new element "CompanyName"


▼Messages 

A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
⋮ [-] e Employee		1	unbounded
[-] ... sequence		1	1
⋮ e EmpNo	string	1	1
⋮ e Dept	string	1	1
⋮ e EmpName	string	1	1
⋮ e Address	string	1	1
⋮ e Tel	string	1	1
⋮ e Salary	string	1	1
⋮ e CompanyName	string	1	1

[Add a Local Element](#)

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".)

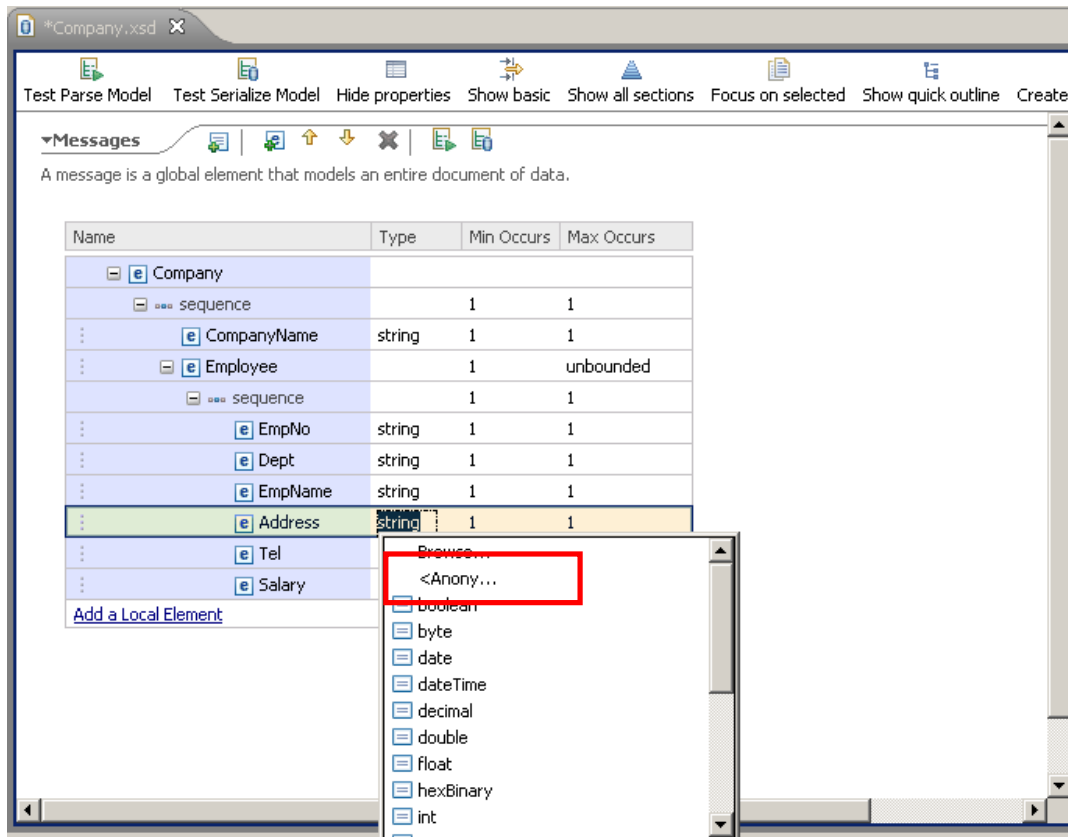
▼Messages 

A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
⋮ [-] e Employee		1	unbounded
[-] ... sequence		1	1
⋮ e EmpNo	string	1	1
⋮ e Dept	string	1	1
⋮ e EmpName	string	1	1
⋮ e Address	string	1	1
⋮ e Tel	string	1	1
⋮ e Salary	string	1	1
⋮ e CompanyName	string	1	1

[Add a Local Element](#)

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 the actual text "Address") and select "Add Sequence".

Name	Type	Min Occurs	Max Occurs
[-] [e] Company			
[-] ... sequence		1	1
⋮ [e] CompanyName	string	1	1
⋮ [-] [e] Employee		1	unbou
[-] ... sequence		1	1
⋮ [e] EmpNo	string	1	1
⋮ [e] Dept	string	1	1
⋮ [e] EmpName	string	1	1
⋮ [-] [e] Address		1	1
⋮			
⋮			
⋮			

Add a Local Element	Make Local Element Global	Alt+Shift+E
	Move to a New Model Group...	Alt+Shift+G
	Move Up	Alt+Up
	Move Down	Alt+Down
	Cut	Ctrl+X
	Copy	Ctrl+C
	Paste	Ctrl+V
	Delete	Delete
	<b>... Add Sequence</b>	<b>Ctrl+L, S</b>
	+ Add Choice	Ctrl+L, C

This will be added like this:

Name	Type	Min Occurs	Max Occurs
[-] [e] Company			
[-] ... sequence		1	1
⋮ [e] CompanyName	string	1	1
⋮ [-] [e] Employee		1	unbounded
[-] ... sequence		1	1
⋮ [e] EmpNo	string	1	1
⋮ [e] Dept	string	1	1
⋮ [e] EmpName	string	1	1
⋮ [-] [e] Address		1	1
[-] ... sequence		1	1
⋮ [e] Tel	string	1	1
⋮ [e] Salary	string	1	1

Add a Local Element

- 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.

Name	Type	Min Occurs	Max Occurs	Default Value
[-] [e] Company				
[-] *** sequence		1	1	
⋮ [e] CompanyName	string	1	1	
⋮ [-] [e] Employee		1	unbounded	
[-] *** sequence		1	1	
⋮ [e] EmpNo	string	1	1	
⋮ [e] Dept	string	1	1	
⋮ [e] EmpName	string	1	1	
⋮ [-] [e] Address		1	1	
[-] *** sequence				
⋮ [e] Tel				
⋮ [e] Salary				
<a href="#">Add a Local Element</a>				

Move Up

Move Down

---

Paste

✕ Delete

---

**+** Add a Local Element

**+** Add Complex Local Element...

\*\*\* Add Sequence

**+** Add Choice

**+** Add Element Reference...

**+** Add Group Reference...

Add Hidden Group Reference... (not supported in current version)

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



▼Messages

A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
⋮ e CompanyName	string	1	1
⋮ [-] e Employee		1	unbounded
[-] ... sequence		1	1
⋮ e EmpNo	string	1	1
⋮ e Dept	string	1	1
⋮ e EmpName	string	1	1
⋮ [-] e Address		1	1
[-] ... sequence		1	1
⋮ e field1	string	1	1
⋮ e field2	string	1	1
⋮ e field3	string	1	1
⋮ e Tel	string	1	1
⋮ e Salary	string	1	1
<a href="#">Add a Local Element</a>			

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

\*Company.xsd

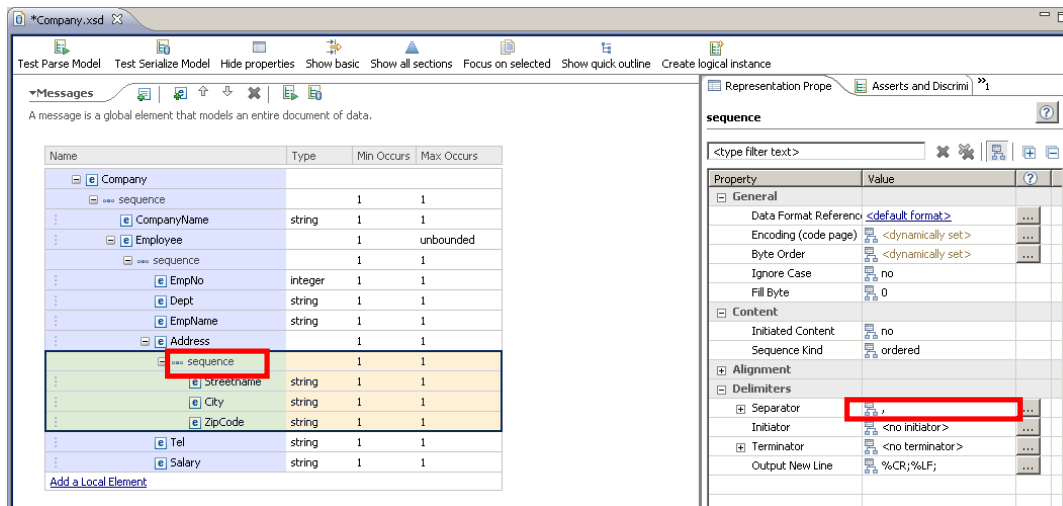
Test Parse Model Test Serialize Model Hide properties Show basic Show all sections Focus on selected Show quick outline

▼Messages

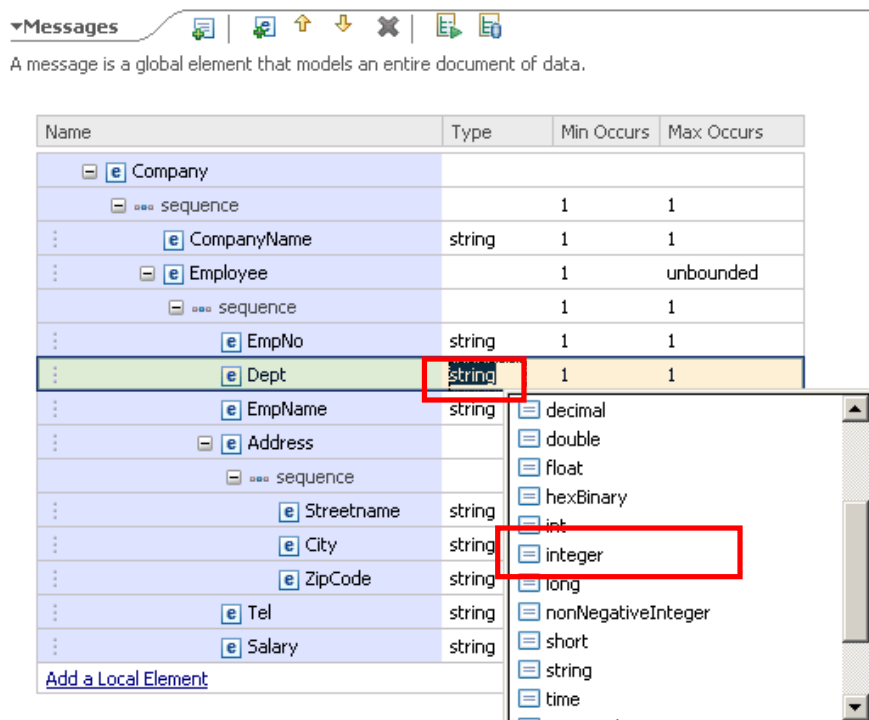
A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
⋮ e CompanyName	string	1	1
⋮ [-] e Employee		1	unbounded
[-] ... sequence		1	1
⋮ e EmpNo	string	1	1
⋮ e Dept	string	1	1
⋮ e EmpName	string	1	1
⋮ [-] e Address		1	1
[-] ... sequence		1	1
⋮ e Streetname	string	1	1
⋮ e City	string	1	1
⋮ e ZipCode	string	1	1
⋮ e Tel	string	1	1
⋮ e Salary	string	1	1
<a href="#">Add a Local Element</a>			

- 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)



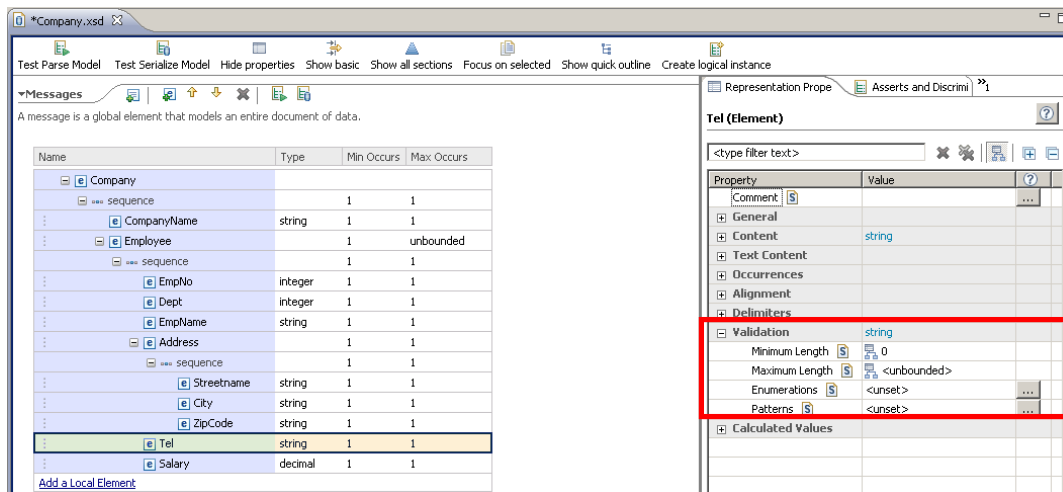
- Click on the type column of the "EmpNo" element and select "integer" (not "int").



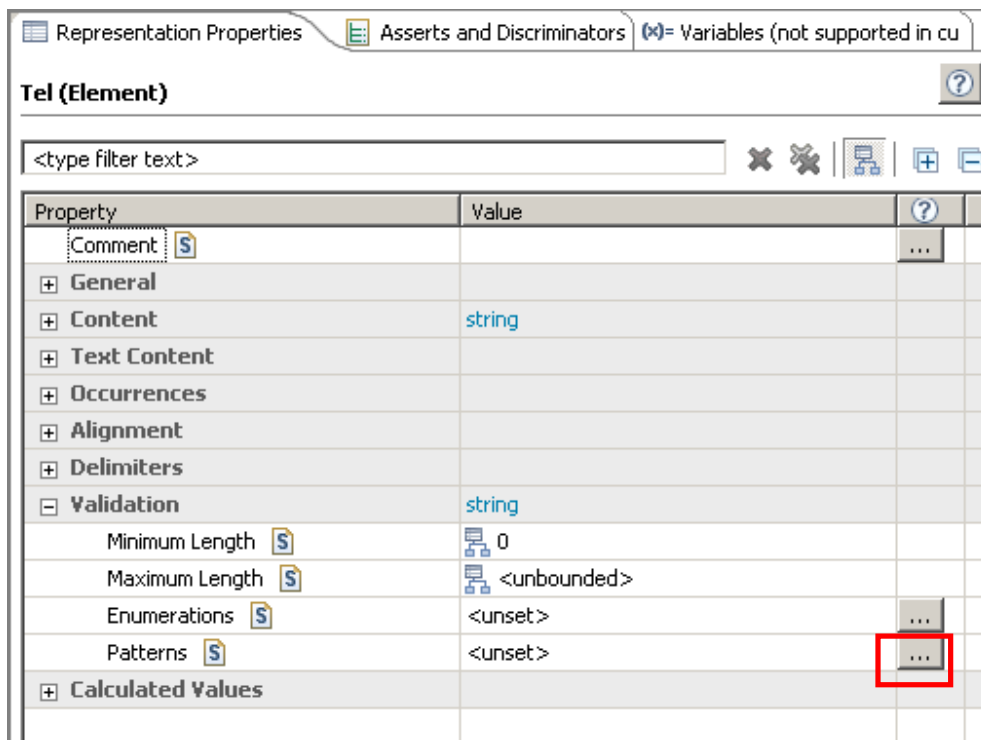
- Similarly, set the Type of the "Dept" element. = integer (not "int", which would restrict the value to 4 bytes).

Set the Type of "Salary" = decimal.

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



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



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

The screenshot shows the 'Property Wizard' dialog box with the 'Pattern' section selected. The dialog has a title bar with 'Property Wizard' and standard window controls. The main area is titled 'Pattern' and contains the following elements:

- A text box for 'Effective pattern (more...):'.
- A 'Test Pattern' section with an 'Example text:' input field and a 'Does example match pattern?' output area.
- A 'Define Pattern' section with two input fields: 'Part of pattern inherited from parent chain:' and 'Part of pattern defined on this type:'.
- A large empty list box for defining pattern parts.
- A vertical stack of buttons on the right side of the list box: 'Add...' (highlighted with a red box), 'Remove', 'Up', and 'Down'.
- A help icon (?) and 'Finish' and 'Cancel' buttons at the bottom.

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

**Regular Expression Wizard**

**Compose Regular Expression**

To add a token, specify its contents and occurrence, then click Add.

Token contents: **Any character (.)**  Auto escape

Occurrence

- Just once
- Zero or more
- One or more
- Optional (?)
- Repeat
- Range  to

**Any character (.)**  
Alphanumeric character (\w)  
Whitespace (\s)  
**Digit (\d)**  
Upper case ([A-Z])  
Lower case ([a-z])  
Current selection

**Add**

Current regular expression:

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

**Regular Expression Wizard**

**Compose Regular Expression**

To add a token, specify its contents and occurrence, then click Add.

Token contents:   Auto escape

Occurrence

Just once

Zero or more ( \* )

One or more ( + )

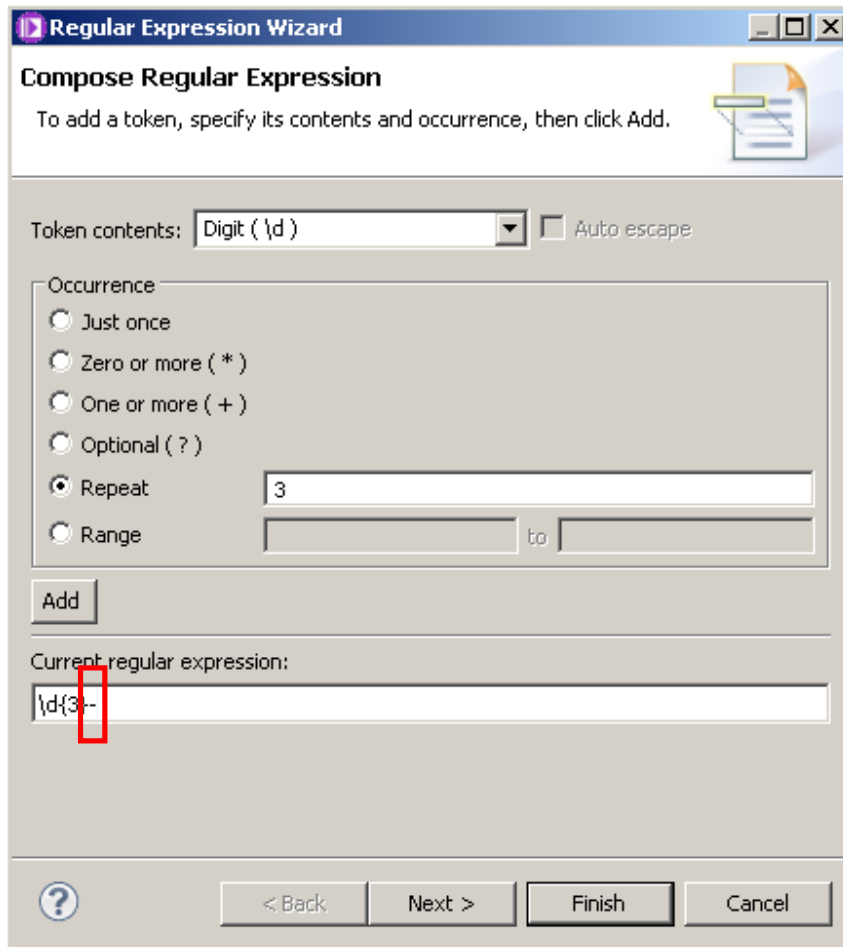
Optional ( ? )

Repeat

Range  to

Current regular expression:

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



**Regular Expression Wizard**

**Compose Regular Expression**

To add a token, specify its contents and occurrence, then click Add.

Token contents:   Auto escape

Occurrence

Just once

Zero or more ( \* )

One or more ( + )

Optional ( ? )

Repeat

Range  to

Current regular expression:

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

**Regular Expression Wizard**

**Compose Regular Expression**

To add a token, specify its contents and occurrence, then click Add.

Token contents:   Auto escape

Occurrence

Just once

Zero or more ( \* )

One or more ( + )

Optional ( ? )

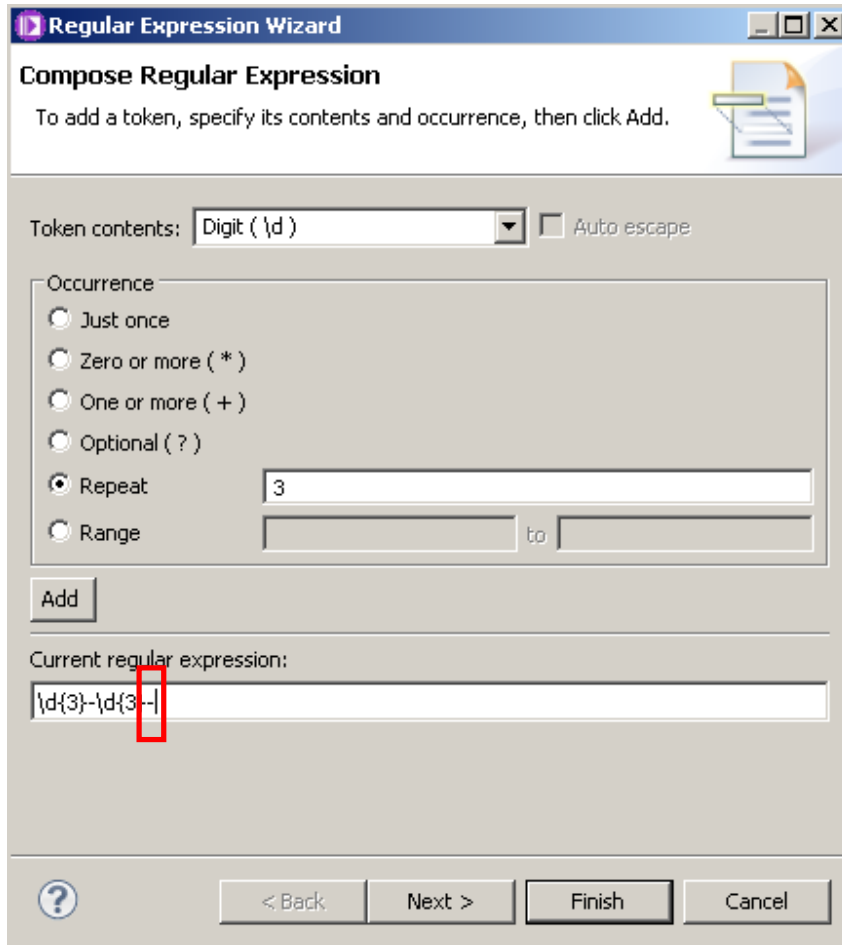
Repeat

Range  to

Current regular expression:



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



**Regular Expression Wizard**

**Compose Regular Expression**

To add a token, specify its contents and occurrence, then click Add.

Token contents:   Auto escape

Occurrence

Just once

Zero or more ( \* )

One or more ( + )

Optional ( ? )

Repeat

Range  to

Current regular expression:

23. 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

**Regular Expression Wizard**

**Compose Regular Expression**

To add a token, specify its contents and occurrence, then click Add.

Token contents:   Auto escape

Occurrence

Just once

Zero or more ( \* )

One or more ( + )

Optional ( ? )

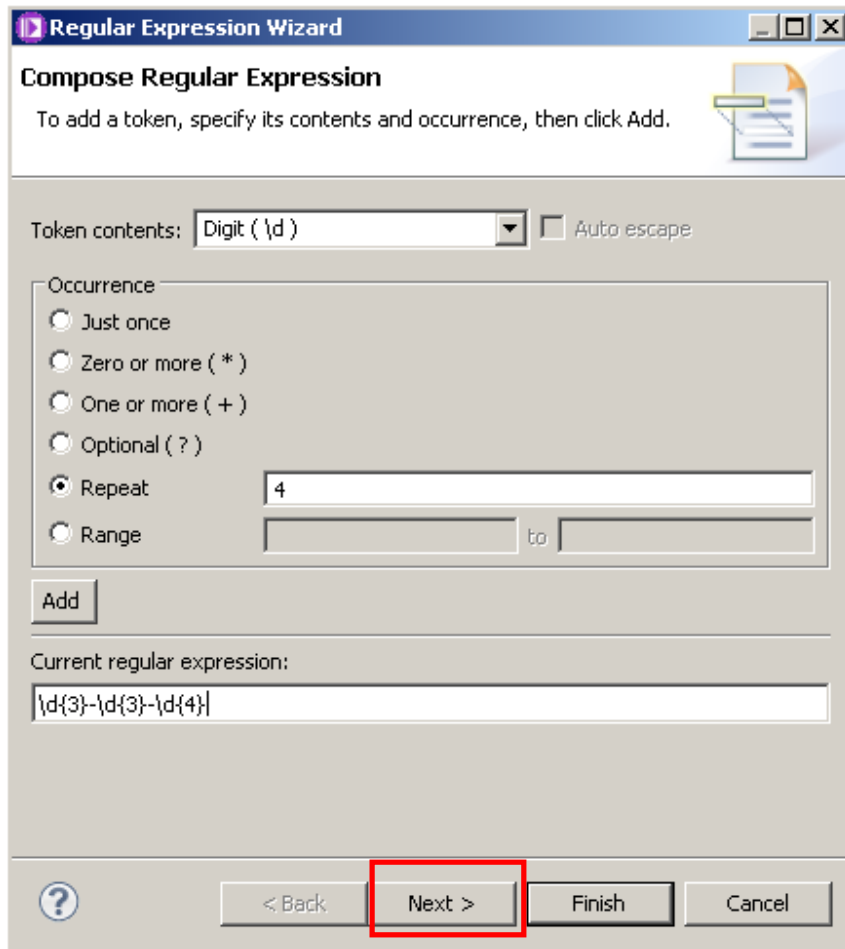
Repeat   to

Range

**Add**

Current regular expression:

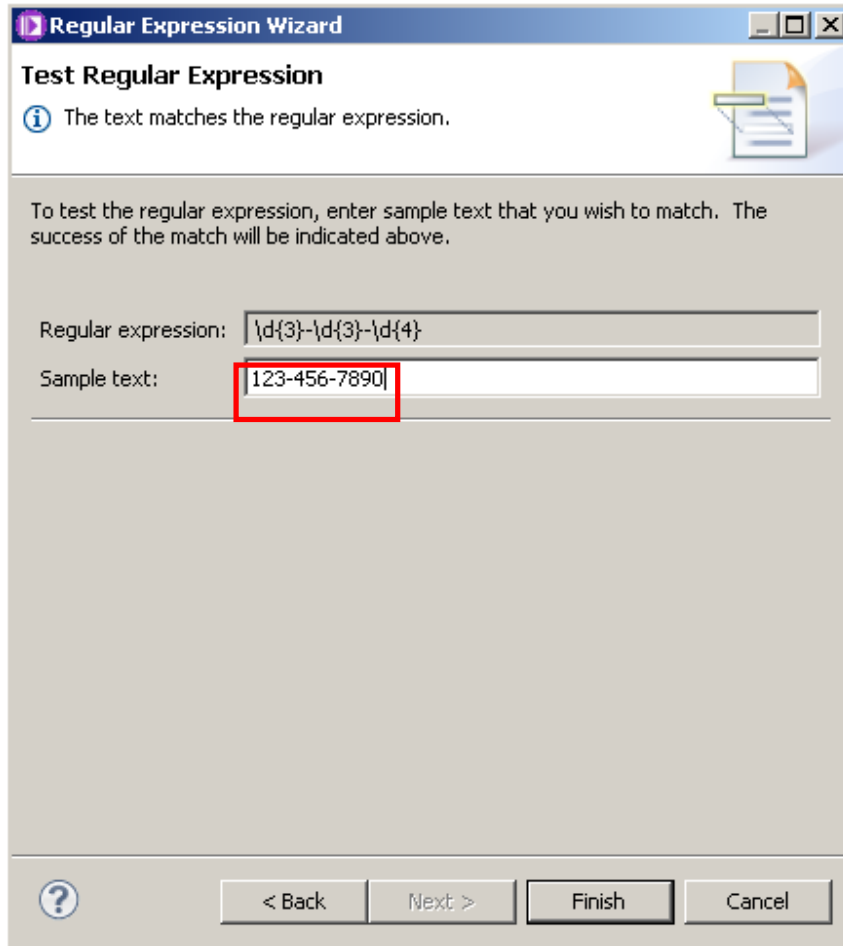
24. Click on the Next button.



The image shows a 'Regular Expression Wizard' dialog box with the following elements:

- Title Bar:** Regular Expression Wizard
- Section Header:** Compose Regular Expression
- Instruction:** To add a token, specify its contents and occurrence, then click Add.
- Token contents:** A dropdown menu showing 'Digit ( \d )' and an 'Auto escape' checkbox.
- Occurrence:** A group box containing radio buttons for 'Just once', 'Zero or more ( \* )', 'One or more ( + )', 'Optional ( ? )', 'Repeat', and 'Range'. The 'Repeat' option is selected, and a text box next to it contains the number '4'.
- Add:** A button to add the current configuration.
- Current regular expression:** A text box containing the expression `\d{3}-\d{3}-\d{4}`.
- Navigation:** A row of buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a red rectangle.

25. 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.

26. 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.

The screenshot shows the 'Property Wizard' dialog box with the 'Pattern' tab selected. The dialog has a title bar with a question mark icon and standard window controls. The main content area is divided into several sections:

- Pattern**: A section with a title and a description: "Set the overall effective pattern by adding parts to the pattern. The pattern can be tested by typing example text. Note that patterns defined here must follow XML Schema regular expression rules." Below this is a text field labeled "Effective pattern (more...)" containing the regular expression `\d{3}-\d{3}-\d{4}`.
- Test Pattern**: A section containing an "Example text:" input field and a "Does example match pattern?" checkbox.
- Define Pattern**: A section with two text fields: "Part of pattern inherited from parent chain:" (empty) and "Part of pattern defined on this type:" containing `\d{3}-\d{3}-\d{4}`. Below these is a list box containing the same pattern `\d{3}-\d{3}-\d{4}`. To the right of the list box are four buttons: "Add...", "Remove", "Up", and "Down".

At the bottom of the dialog, there is a help icon (question mark) on the left and "Finish" and "Cancel" buttons on the right.

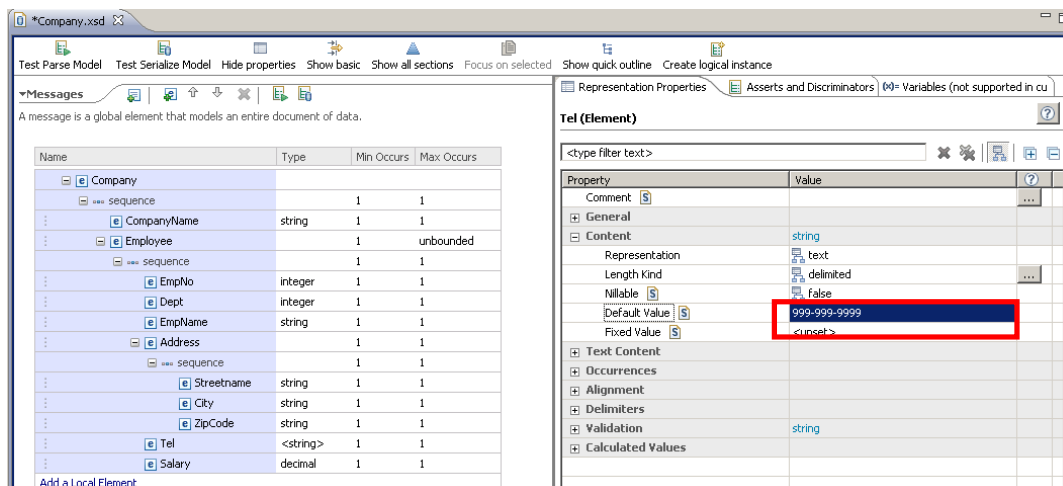
- Notice that the "Tel" element's type has changed from "string" to "<string>", an anonymous local restriction of xs:string, in order to carry the pattern facet.

**Messages**   
 A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
[-] e Company			
[-] ... sequence		1	1
...			
e CompanyName	string	1	1
...			
[-] e Employee		1	unbounded
[-] ... sequence		1	1
...			
e EmpNo	integer	1	1
...			
e Dept	integer	1	1
...			
e EmpName	string	1	1
...			
[-] e Address		1	1
[-] ... sequence		1	1
...			
e Streetname	string	1	1
...			
e City	string	1	1
...			
e ZipCode	string	1	1
...			
e Tel	<string>	1	1
...			
e Salary	decimal	1	1

[Add a Local Element](#)

- 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).



The screenshot shows the software interface for editing the 'Tel' element. On the left, a tree view shows the element's structure. On the right, the 'Representation Properties' pane is open for the 'Tel (Element)'. The 'Default Value' property is set to '999-999-9999', which is highlighted with a red box. Other properties like 'Content', 'Length Kind', and 'Validation' are also visible.

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

Expand Text Number Representation.

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

The screenshot shows the DFDL Editor interface. On the left, the 'Messages' pane displays a tree view of a message structure. The 'Salary' element is highlighted in blue. On the right, the 'Representation Properties' pane for the 'Salary (Element)' is shown. The 'Text Content' section is expanded, and the 'Text Number Representation' sub-section is also expanded. The 'Number Pattern' property is highlighted with a red box, and its value is '#0.###'. The 'Number Pattern' property has a three-dot menu button next to it.

Name	Type	Min Occurs	Max Occurs
Company			
sequence		1	1
CompanyName	string	1	1
Employee		1	unbounded
sequence		1	1
EmpNo	integer	1	1
Dept	integer	1	1
EmpName	string	1	1
Address		1	1
sequence		1	1
Streetname	string	1	1
City	string	1	1
ZipCode	string	1	1
Tel	<string>	1	1
Salary	decimal	1	1

Property	Value
Comment	S
General	
Text Content	decimal
Text Number Representation	standard
Number Base	10
Text Number Check Policy	lax
Number Pattern	#0.###
Grouping Separator	,
Decimal Separator	.
Exponent Character	E
Zero Representation	
Rounding	pattern
Number Justification	right
Number Pad Character	%SP;
Decimal Signed	yes

30. 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.



The screenshot shows the 'Property Wizard' dialog box for configuring a 'Number Pattern'. The title bar reads 'Property Wizard'. The main heading is 'Number Pattern' with the subtitle 'Set and test values for the number pattern properties.' Below this, a note states: 'Pattern describes the format of the text number. Click [here](#) to see symbols and meanings.'

The 'Pattern' dropdown menu is set to '#0.##', which is highlighted with a red box. Below this, the 'Text Format' section is visible. It includes a 'Number Type' dropdown set to 'decimal'. The 'Number' input field contains '1234.1234', and the 'Formatted' field is empty. The 'Apply Pattern' button is highlighted with a red box, and the 'Extract Pattern' button is also visible.

The 'Standard number' radio button is selected. It is accompanied by a description: 'A number is represented as characters in the 'encoding' code page. [More...](#)'. Below this are several configuration options:

- Number base: 10
- Number check policy: lax
- Grouping separator: ,
- Decimal separator: .
- Exponent character: E
- Infinity representation character: Inf
- NaN representation character: NaN
- Zero representation character: (empty)
- Number rounding: pattern
- Number rounding increment: (empty)
- Number rounding mode: roundUp

The 'Zoned number' radio button is unselected, with a description: 'A number is represented as zoned decimal in the 'encoding' code page. [More...](#)'. At the bottom of the dialog, the 'Finish' button is highlighted with a red box, and the 'Cancel' button is also visible.

Notice that the number was changed from "1234.1234" to "1234.12" to comply with the defined number pattern.

Click Finish.

31. 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.

The screenshot shows the DFDL Editor interface. On the left, a tree view displays the 'Company' element and its sub-elements: 'sequence', 'CompanyName', 'Employee', 'sequence', 'EmpNo', 'Dept', 'EmpName', 'Address', 'sequence', 'Streetname', 'City', 'ZipCode', 'Tel', and 'Salary'. On the right, the 'Representation Properties' view for the 'Company (Element)' is shown. The 'Delimiters' section is highlighted with a red box, showing the following properties:

Property	Value
Initiator	<no initiator>
Terminator	<no terminator>
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;

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

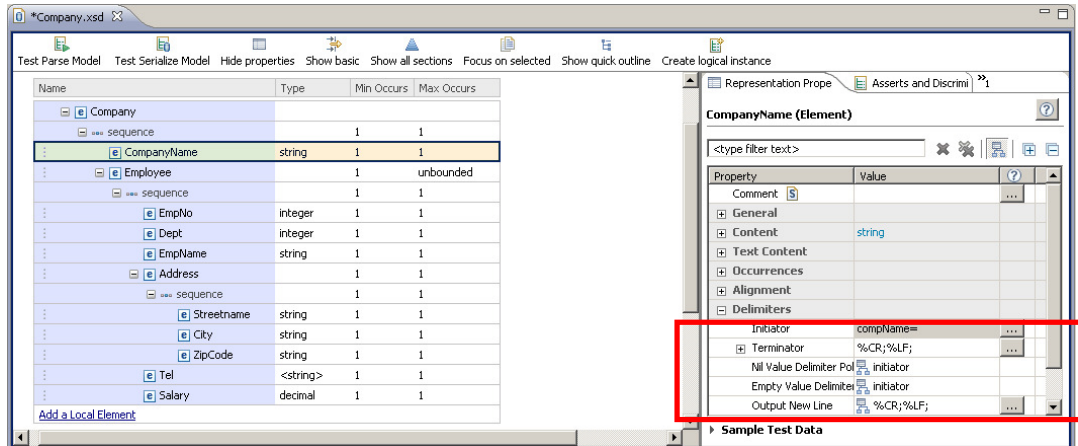
Hint: after you have entered the "]", you can use Ctrl-Space to use the Toolkit Content Assist editor, and select the CR and LF values.

The screenshot shows the DFDL Editor interface. On the left, the tree view is the same as in the previous screenshot. On the right, the 'Representation Properties' view for the 'Company (Element)' is shown. The 'Delimiters' section is highlighted with a red box, showing the following updated properties:

Property	Value
Initiator	Company[
Terminator	]%CR;%LF;
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;

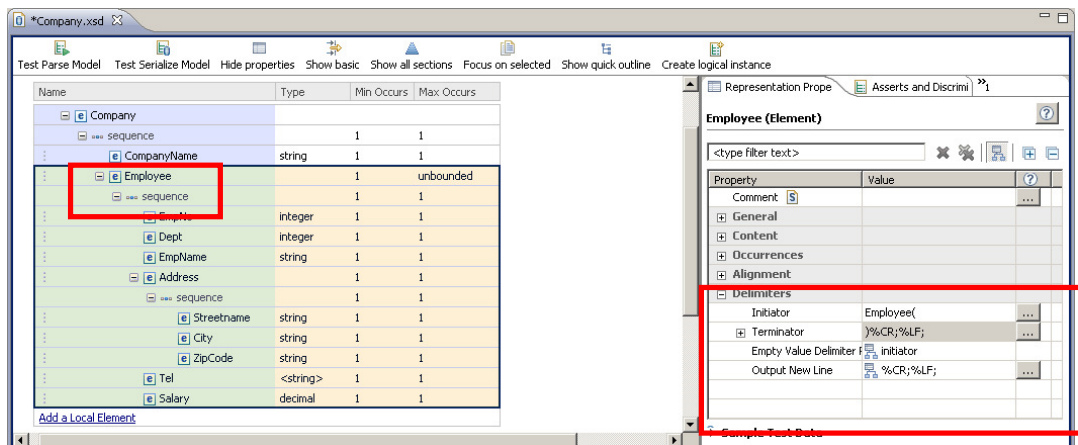
This definition implies that the record starts with a "Company[" tag and ends with a "]%CR;%LF;" tag.

- 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:

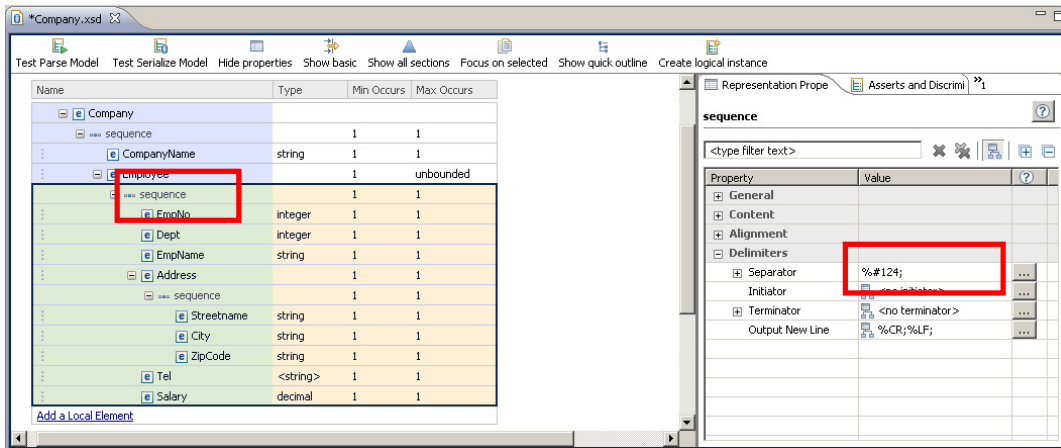


- Click on the "Employee" element, and in the "Delimiter" section of the Representation properties view, set the Terminator value to ")%CR;%LF;". Make sure you don't miss the ")" at the start of the terminator string.

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



- 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)

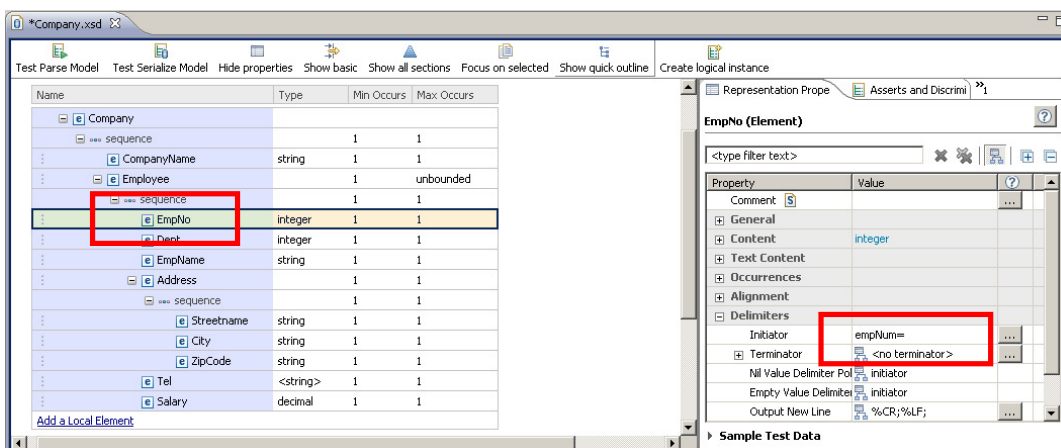


This Separator defines that all the fields inside the "Employee" structure are separated by the "|" character.

- 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".



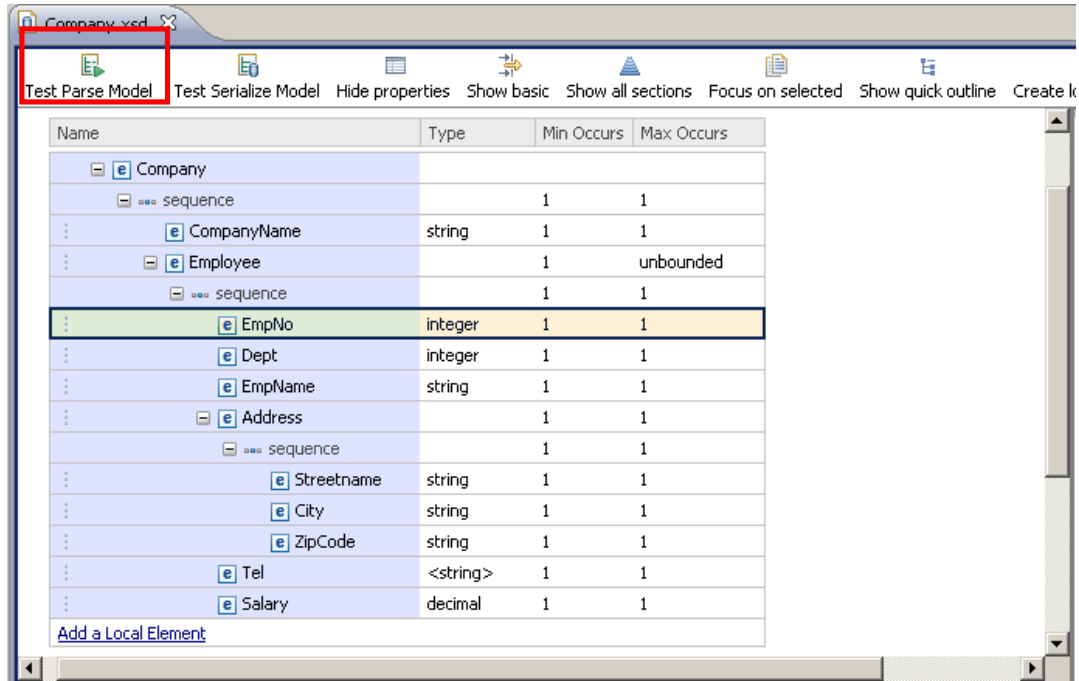
- Save your DFDL Schema by pressing Ctrl+S or File->Save. When saved, the DFDL Schema is validated and if any errors (or warnings) are found, they will appear in the Problems view.

Make sure there are no errors in the Problems view.

## 4. Testing the Message Model

1. 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

**Test Parse Model**

Message  
Select message for testing. [More...](#)  
Message name: \* Company

Parser Input  
Select content to be parsed against schema.  
 Content from 'DFDL Test - Serialize' view  
 Content from a data file  
Input file name: \*  [Browse...](#)

Specify runtime configuration.

Runtime encoding options  
Provide runtime values for properties which have been configured in the model to be dynamically set. [More...](#)  
Encoding (code page): UTF-8  
Floating point format: IEEE Non-Extended  
Byte order:  Little endian  Big endian

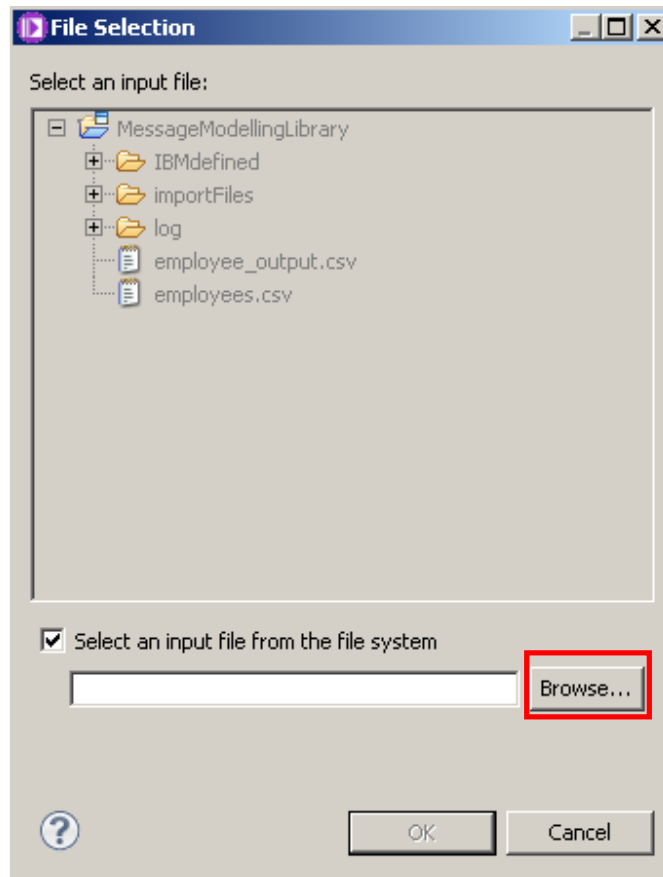
Runtime validation  
 Validate data against schema [More...](#)

[Restore Defaults](#)

OK Cancel

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

Browse to the "C:\student\MessageModeling\data" directory and select "Company.txt".  
Click OK, and then OK again.



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

Click OK.

**Test Parse Model**

Message  
Select message for testing. [More...](#)  
Message name: \* Company

Parser Input  
Select content to be parsed against schema.  
 Content from 'DFDL Test - Serialize' view  
 Content from a data file  
Input file name: \* C:\student\MessageModeling\data\Company.txt [Browse...](#)

Specify runtime configuration.

Runtime encoding options  
Provide runtime values for properties which have been configured in the model to be dynamically set. [More...](#)  
Encoding (code page): UTF-8  
Floating point format: IEEE Non-Extended  
Byte order:  Little endian  Big endian

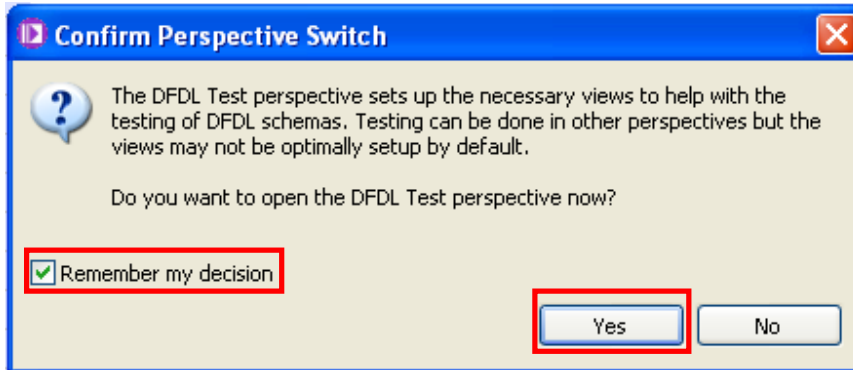
Runtime validation  
 Validate data against schema [More...](#)

[Restore Defaults](#)

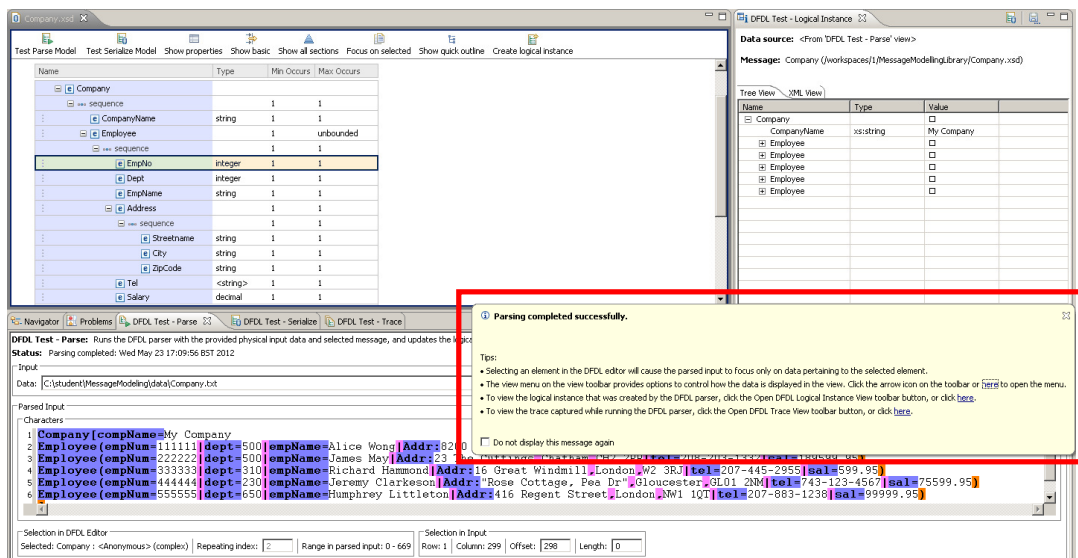
OK Cancel



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

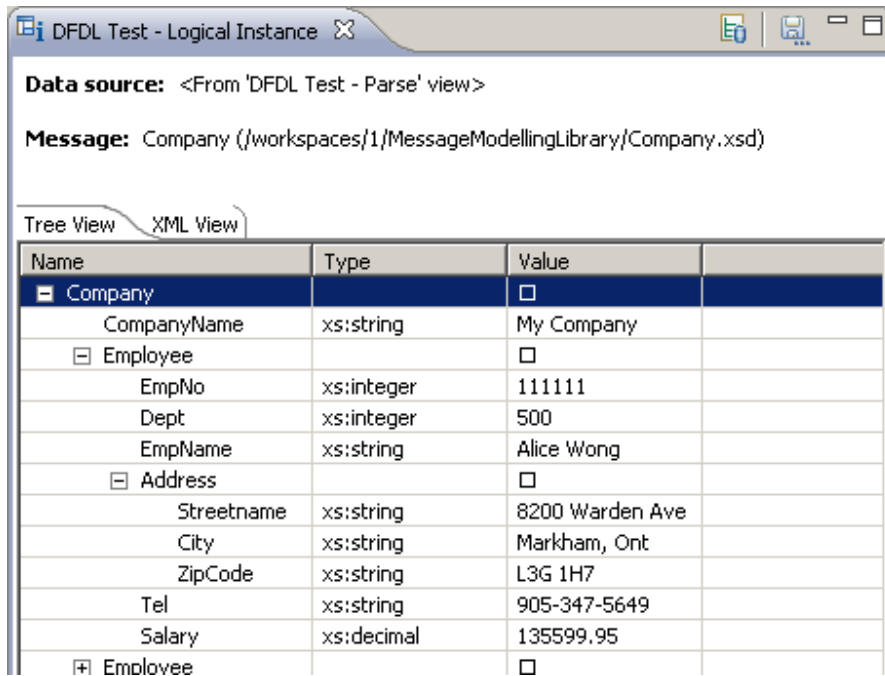


- In the "DFDL Test" perspective, "DFDL Test - Parse" view, a message bubble appears indicating the parsing was successful.



Close the message by clicking on the "X".

7. 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		<input type="checkbox"/>
CompanyName	xs:string	My Company
Employee		<input type="checkbox"/>
EmpNo	xs:integer	111111
Dept	xs:integer	500
EmpName	xs:string	Alice Wong
Address		<input type="checkbox"/>
Streetname	xs:string	8200 Warden Ave
City	xs:string	Markham, Ont
ZipCode	xs:string	L3G 1H7
Tel	xs:string	905-347-5649
Salary	xs:decimal	135599.95
Employee		<input type="checkbox"/>

This concludes the Record-oriented (tagged, delimited) text Message Modeling lab.