16L93

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

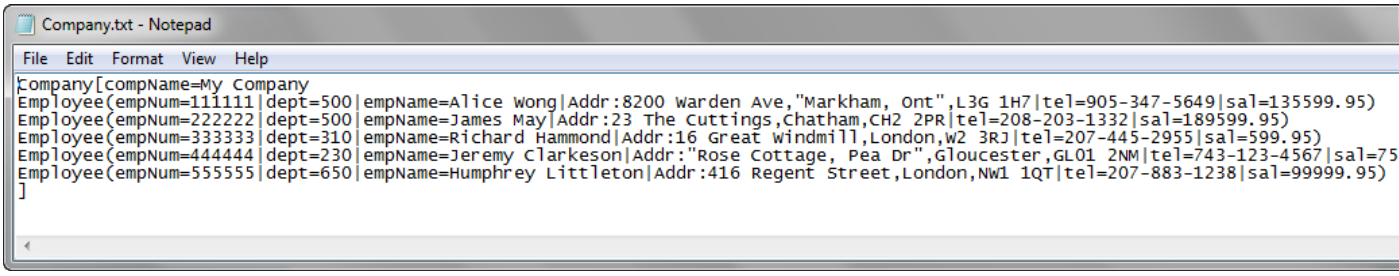
1.1 Lab preparation

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

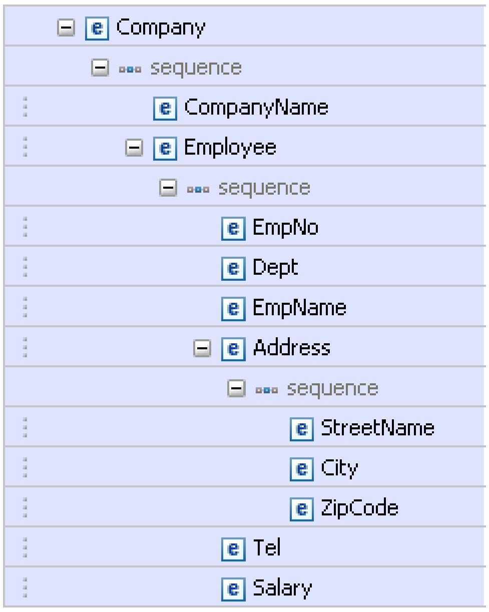
A Record oriented message model is useful to model messages that consist of text strings, but it can also handle binary data. Examples of this type of messages are those that conform to the ACORD AL3, EDIFACT, HL7, SWIFT, or X12 standards. This format allows a high degree of flexibility when defining 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:



```
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 Clarkeson|Addr:"Rose Cottage, Pea Dr",Gloucester,GL01 2NM|tel=743-123-4567|sal=75
Employee(empNum=555555|dept=650|empName=Humphrey Littleton|Addr:416 Regent Street,London,NW1 1QT|tel=207-883-1238|sal=99999.95)
]
```

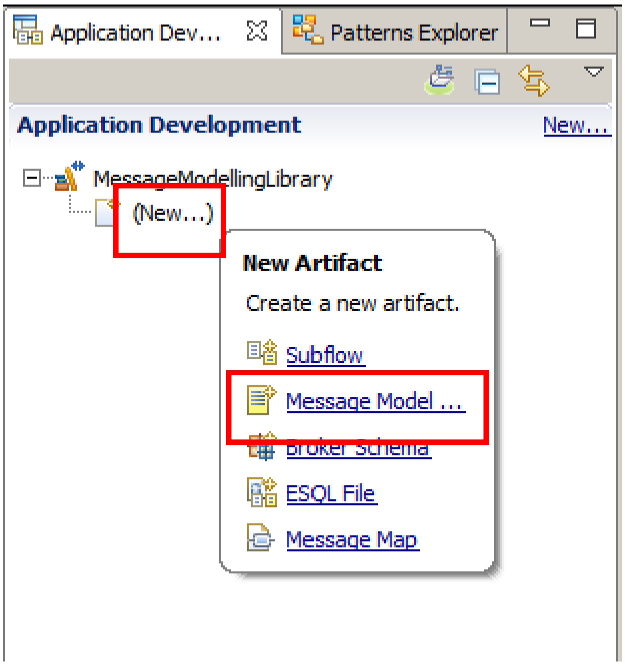
This is an outline of the final message model that you will create:



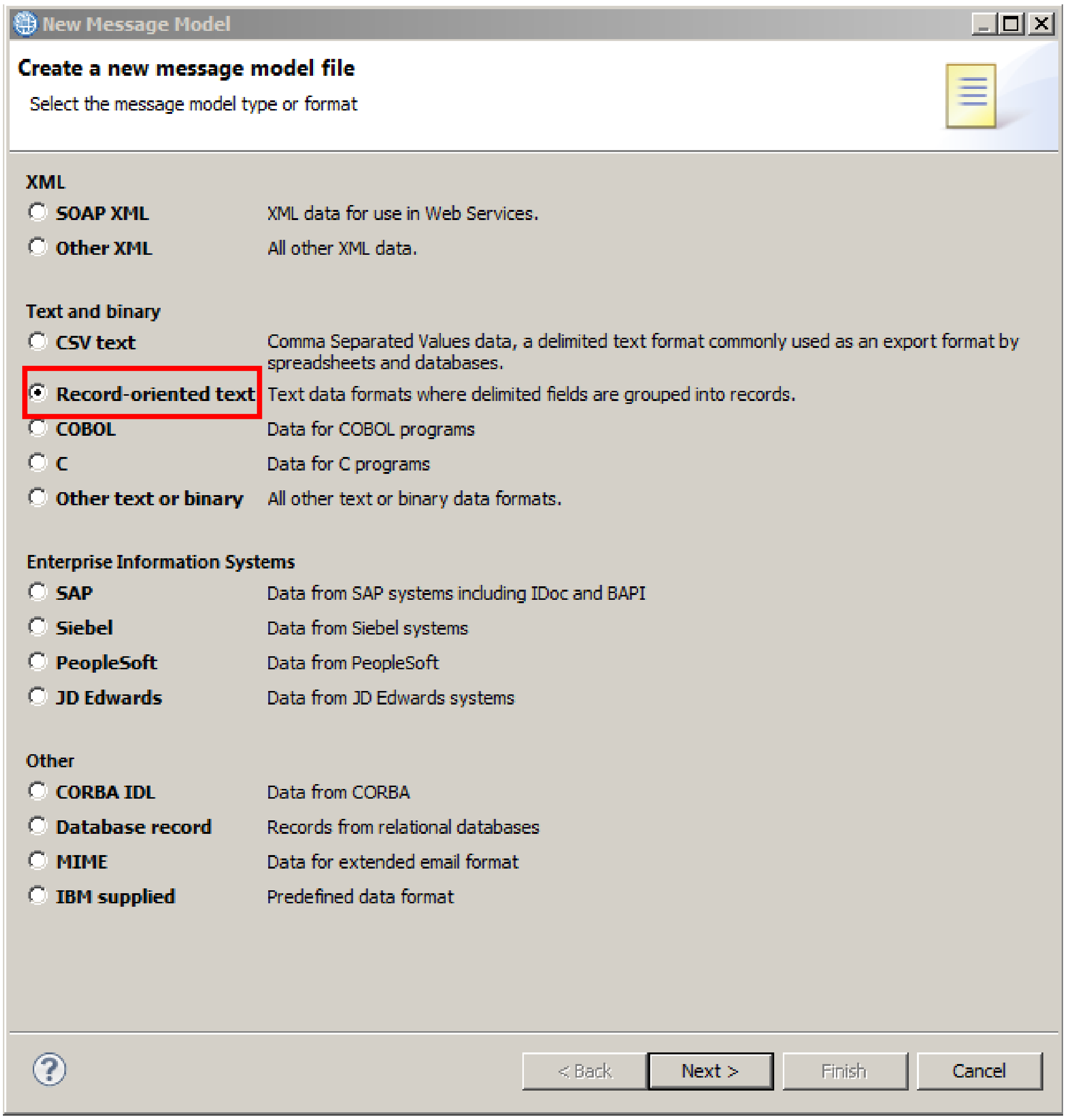
2. Build the Message Model

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

   (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.



Create a new message model file

Select the message model type or format

XML

○ SOAP XML	XML data for use in Web Services.
○ Other XML	All other XML data.

Text and binary

○ CSV text	Comma Separated Values data, a delimited text format commonly used as an export format by spreadsheets and databases.
● Record-oriented text	Text data formats where delimited fields are grouped into records.
○ COBOL	Data for COBOL programs
○ C	Data for C programs
○ Other text or binary	All other text or binary data formats.

Enterprise Information Systems

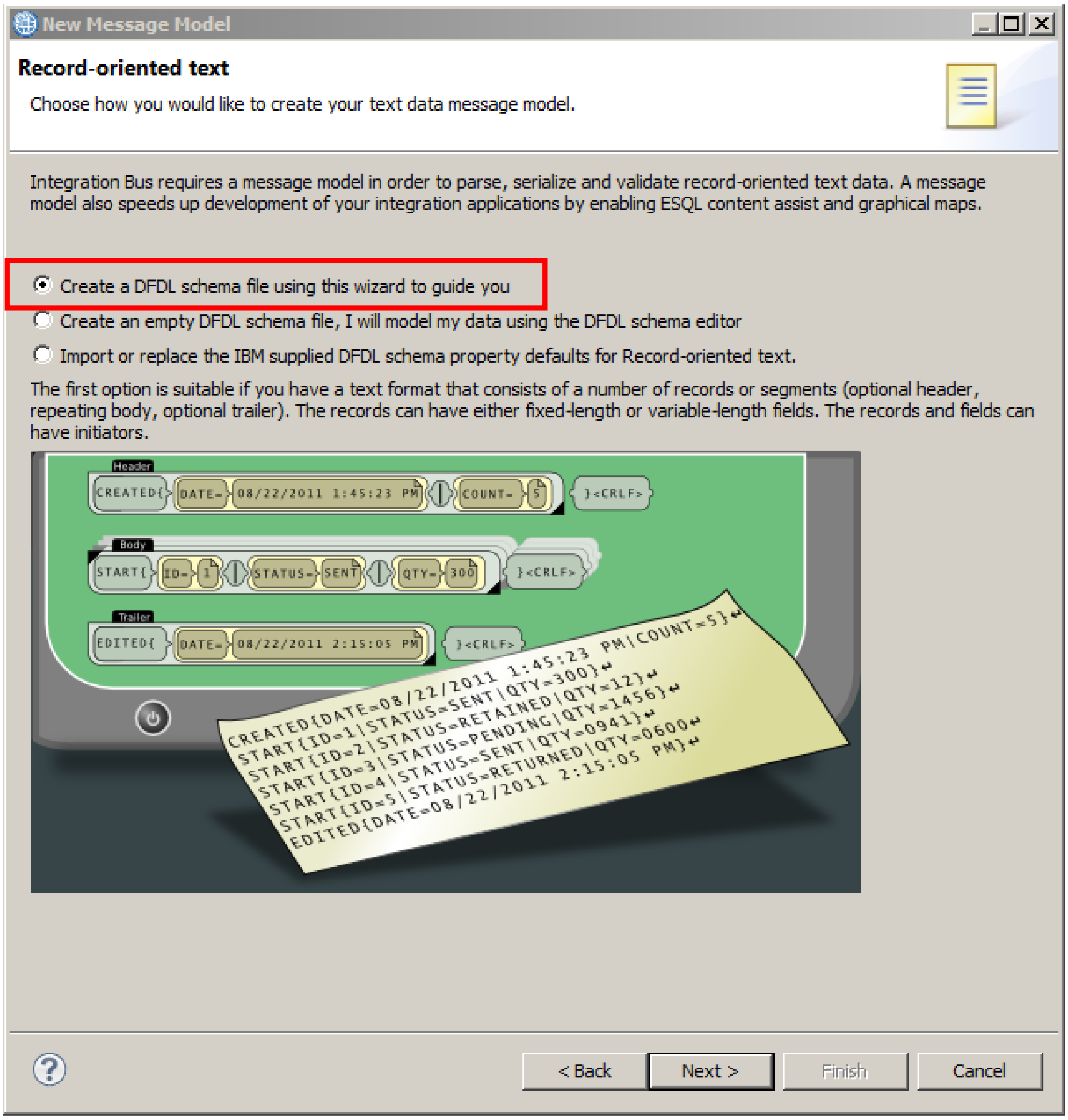
○ SAP	Data from SAP systems including IDoc and BAPI
○ Siebel	Data from Siebel systems
○ PeopleSoft	Data from PeopleSoft
○ JD Edwards	Data from JD Edwards systems

Other

○ CORBA IDL	Data from CORBA
○ Database record	Records from relational databases
○ MIME	Data for extended email format
○ IBM supplied	Predefined data format

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.



Record-oriented text

Choose how you would like to create your text data message model.

Integration Bus requires a message model in order to parse, serialize and validate record-oriented text data. A message model also speeds up development of your integration applications by enabling ESQL content assist and graphical maps.

● Create a DFDL schema file using this wizard to guide you

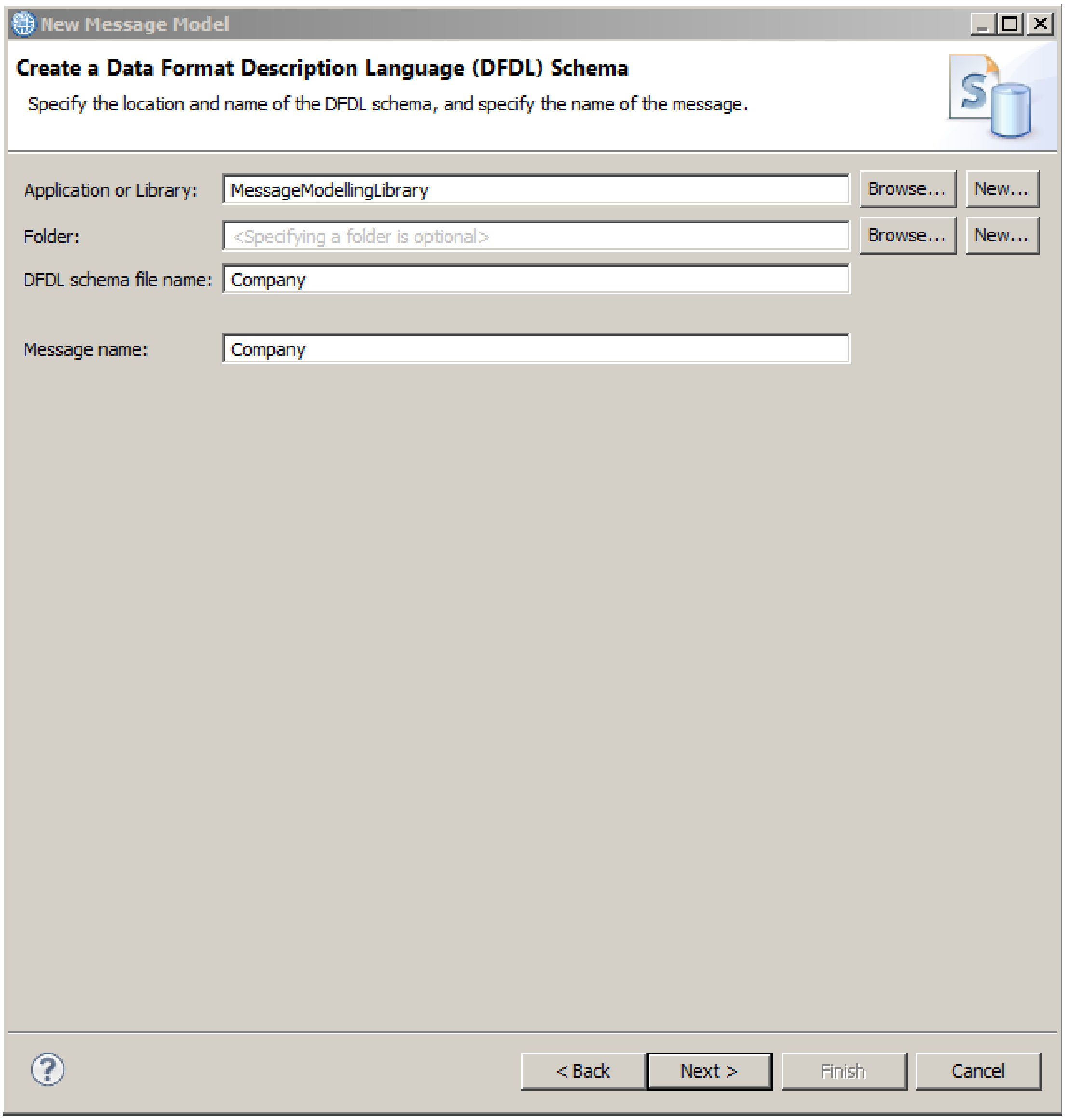
○ Create an empty DFDL schema file, I will model my data using the DFDL schema editor

○ Import or replace the IBM supplied DFDL schema property defaults for Record-oriented text.

The first option is suitable if you have a text format that consists of a number of records or segments (optional header, repeating body, optional trailer). The records can have either fixed-length or variable-length fields. The records and fields can have initiators.

```
CREATED{DATE=08/22/2011 1:45:23 PM|COUNT=5}↵
START{ID=1|STATUS=SENT|QTY=300}↵
START{ID=2|STATUS=RETAINED|QTY=12}↵
START{ID=3|STATUS=PENDING|QTY=1456}↵
START{ID=4|STATUS=SENT|QTY=0941}↵
START{ID=5|STATUS=RETURNED|QTY=0600}↵
EDITED{DATE=08/22/2011 2:15:05 PM}↵
```

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: MessageModellingLibrary

Folder: <Specifying a folder is optional>

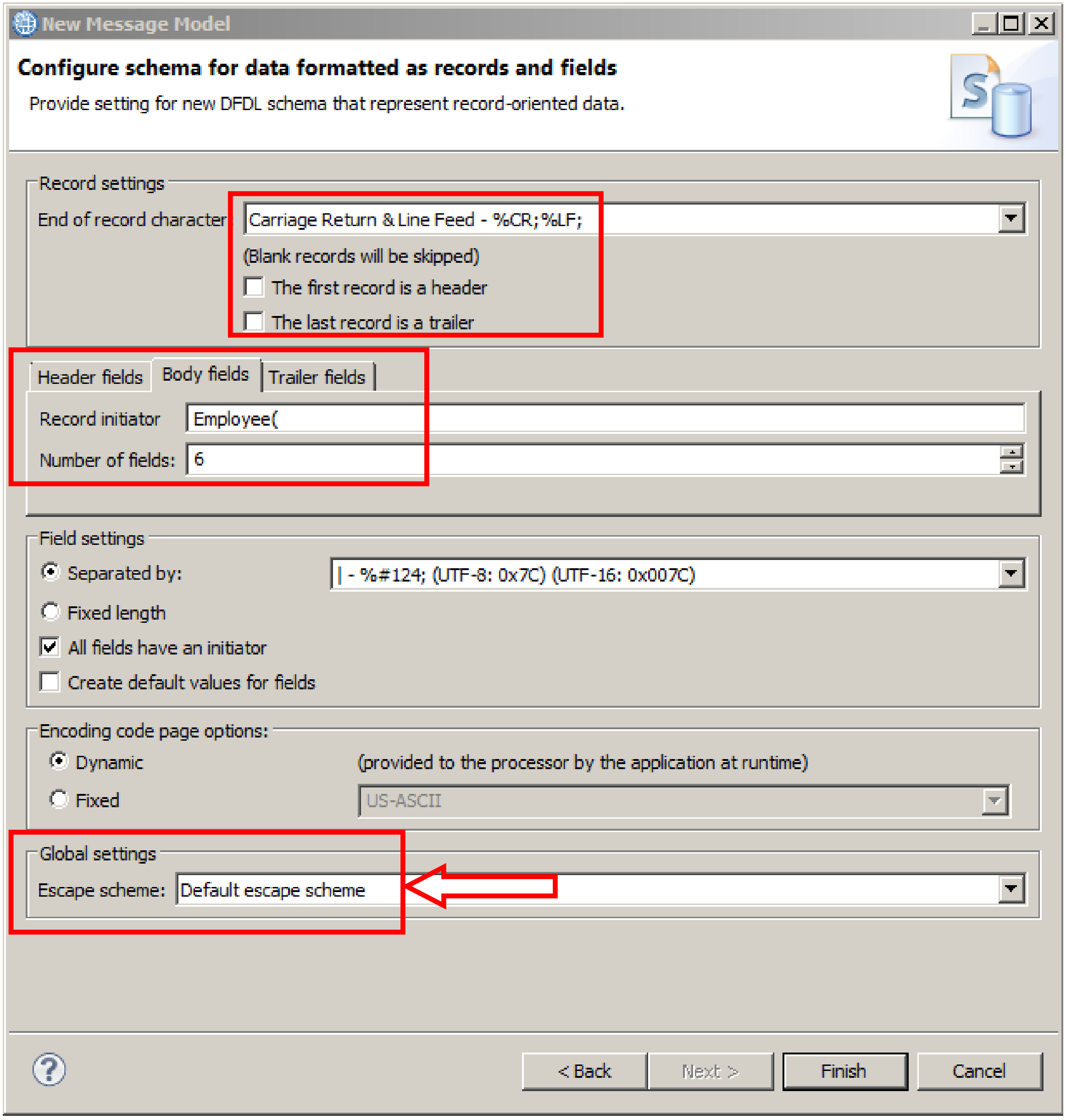
DFDL schema file name: Company

Message name: Company

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.



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 US-ASCII

Global settings

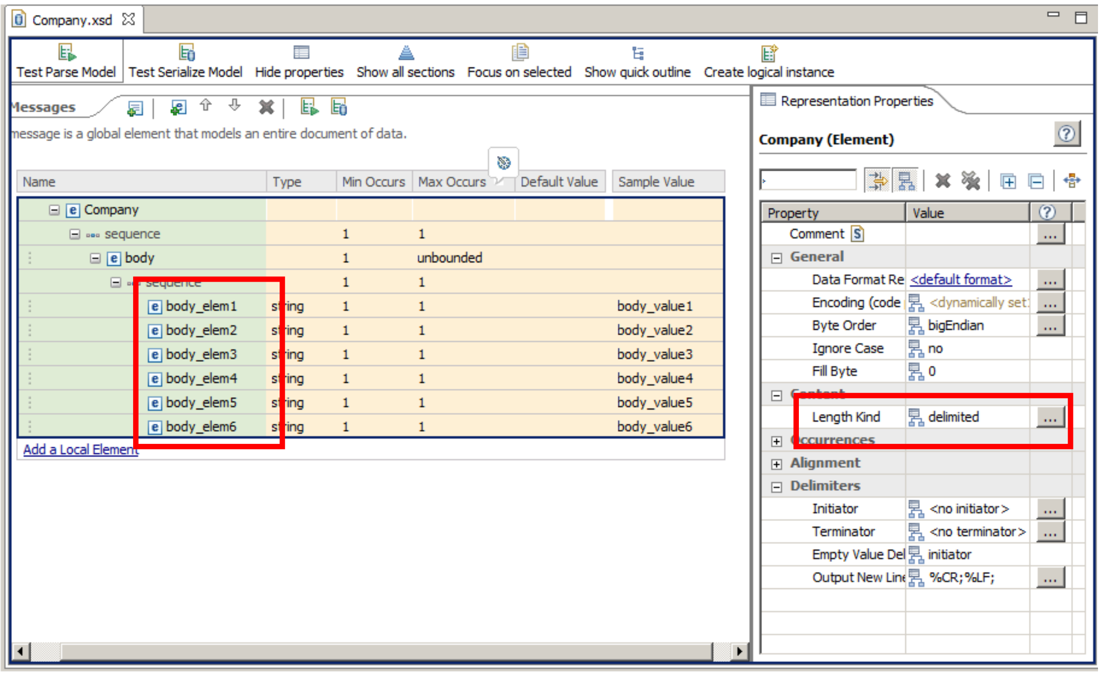
Escape scheme: Default escape scheme

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.



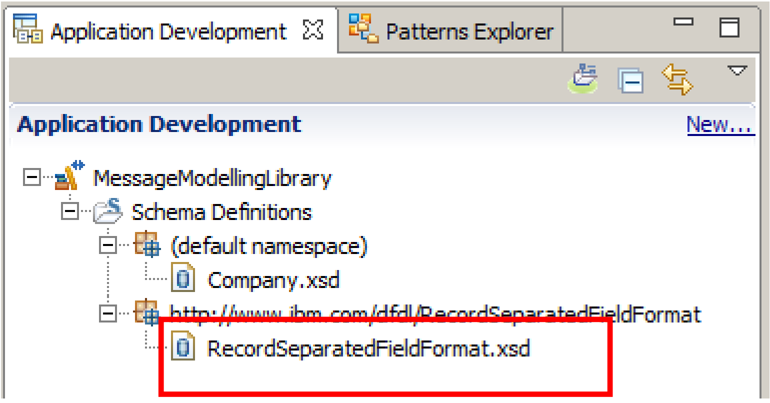
message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
body		1	unbounded		
sequence		1	1		
body_elem1	string	1	1		body_value1
body_elem2	string	1	1		body_value2
body_elem3	string	1	1		body_value3
body_elem4	string	1	1		body_value4
body_elem5	string	1	1		body_value5
body_elem6	string	1	1		body_value6

Company (Element)

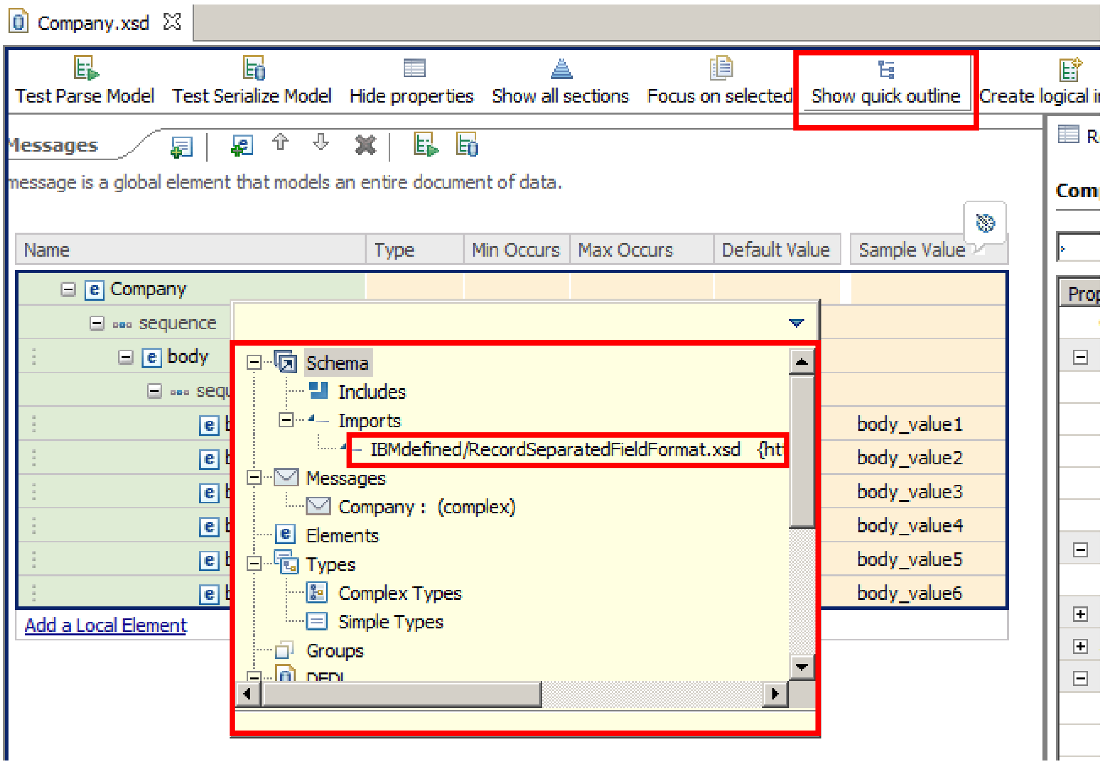
Property	Value
Comment	
General	
Data Format Re	<default format>
Encoding (code	<dynamically set
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	
Length Kind	delimited
Occurrences	
Alignment	
Delimiters	
Initiator	<no initiator>
Terminator	<no terminator>
Empty Value Del	initiator
Output New Line	%CR;%LF;

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.



MessageModellingLibrary
Schema Definitions
(default namespace)
Company.xsd
http://www.ibm.com/dfdl/RecordSeparatedFieldFormat
RecordSeparatedFieldFormat.xsd

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

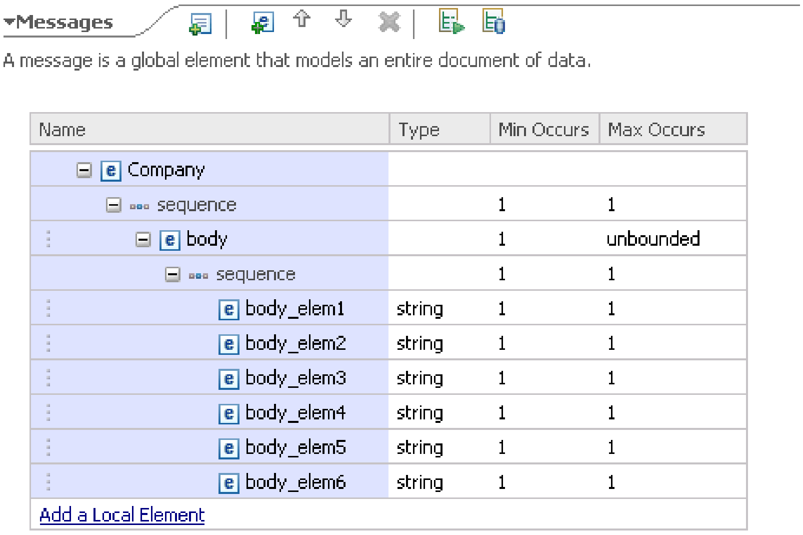


The Outline view will appear with a high level view of the elements of your message model. If you click on any of them, the editor will focus on it.

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.

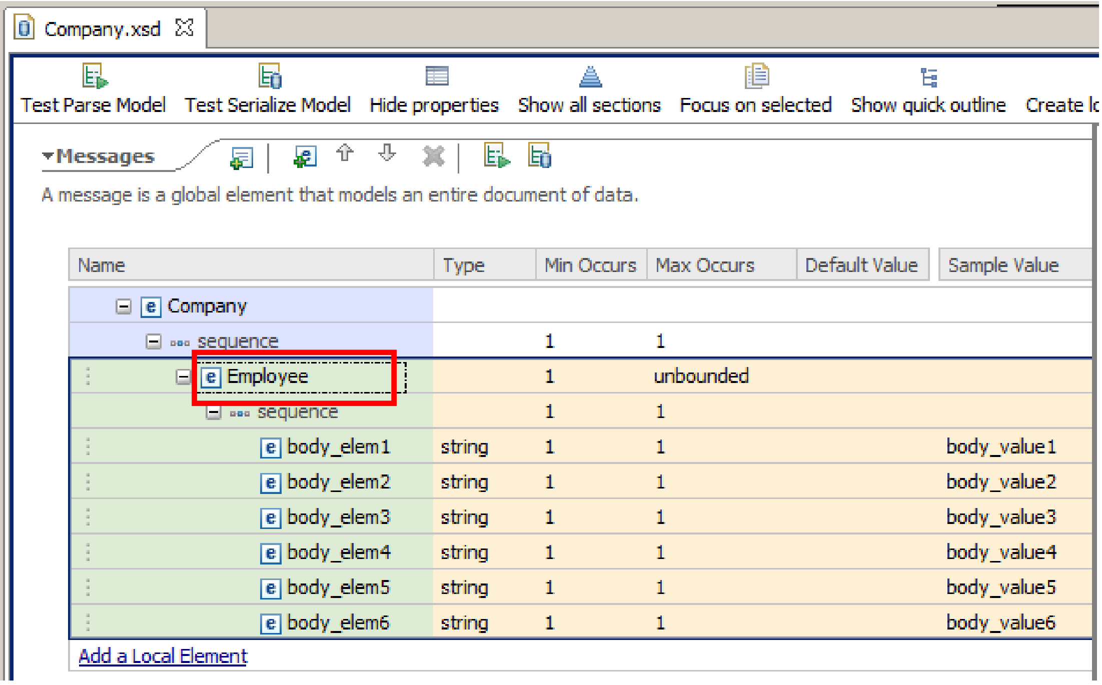


A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
Company			
sequence		1	1
body		1	unbounded
sequence		1	1
body_elem1	string	1	1
body_elem2	string	1	1
body_elem3	string	1	1
body_elem4	string	1	1
body_elem5	string	1	1
body_elem6	string	1	1
Add a Local Element			

3. Refining the Message Model

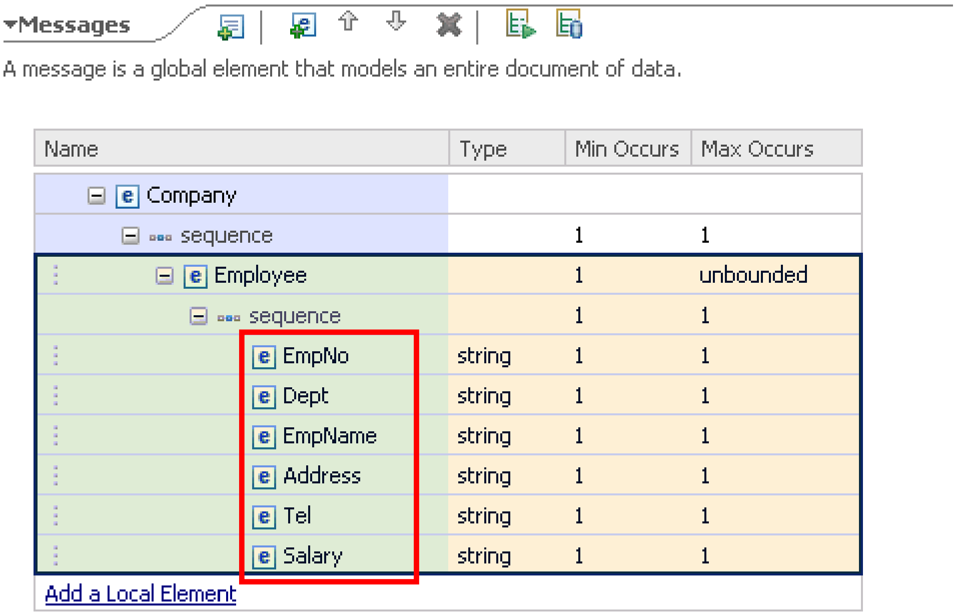
1. Change the name of the "body" field to "Employee" by single-clicking on it, and overtyping.



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

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
Employee		1	unbounded		
sequence		1	1		
body_elem1	string	1	1		body_value1
body_elem2	string	1	1		body_value2
body_elem3	string	1	1		body_value3
body_elem4	string	1	1		body_value4
body_elem5	string	1	1		body_value5
body_elem6	string	1	1		body_value6

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.

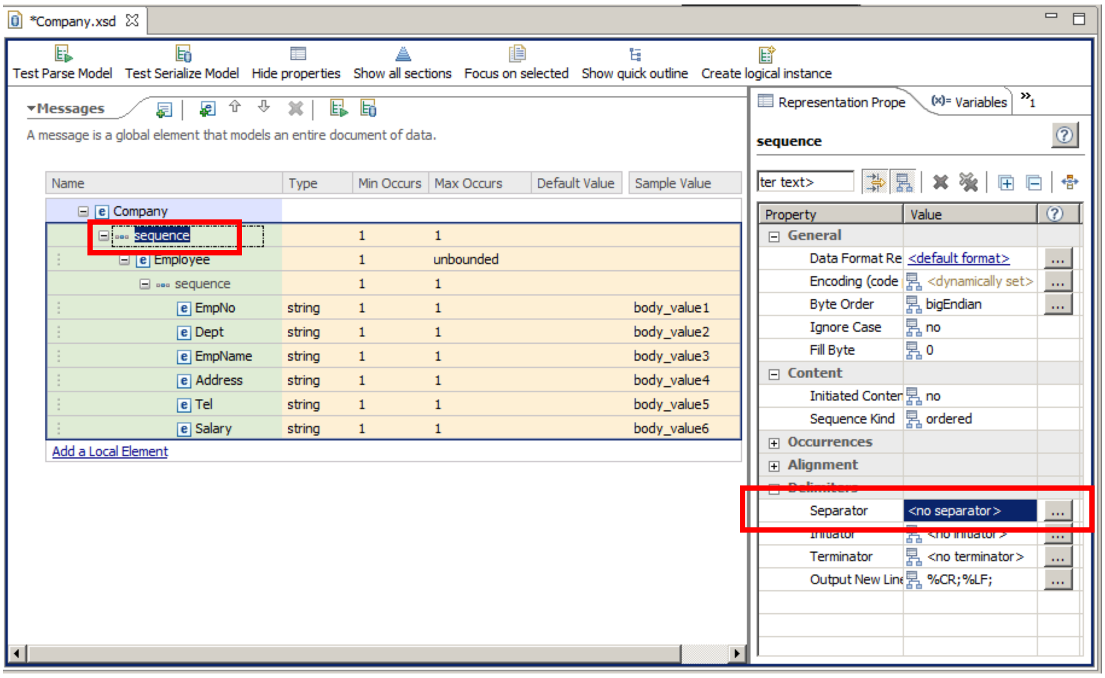


A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs
Company			
sequence		1	1
Employee		1	unbounded
sequence		1	1
EmpNo	string	1	1
Dept	string	1	1
EmpName	string	1	1
Address	string	1	1
Tel	string	1	1
Salary	string	1	1

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).



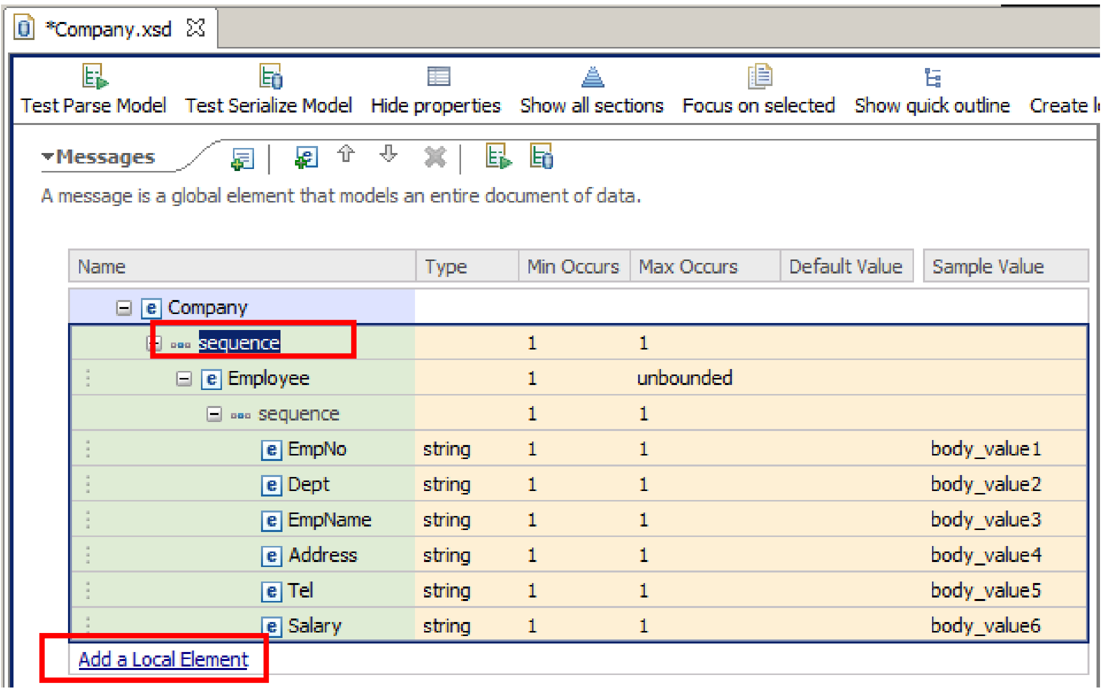
A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
Employee		1	unbounded		
sequence		1	1		
EmpNo	string	1	1		body_value1
Dept	string	1	1		body_value2
EmpName	string	1	1		body_value3
Address	string	1	1		body_value4
Tel	string	1	1		body_value5
Salary	string	1	1		body_value6

sequence

Property	Value
General	
Data Format Re	<default format>
Encoding (code	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	
Initiated Conter	no
Sequence Kind	ordered
Occurrences	
Alignment	
Delimiters	
Separator	<no separator>
Initiator	<no initiator>
Terminator	<no terminator>
Output New Line	%CR;%LF;

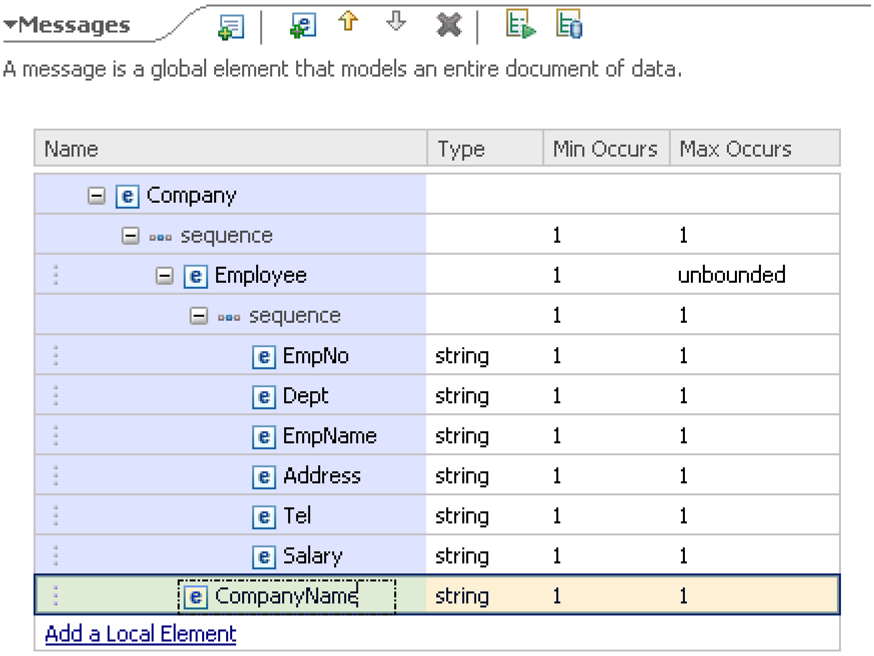
4. Click on the <sequence> content of the Company element and click on "Add a Local Element".



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

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
Employee		1	unbounded		
sequence		1	1		
EmpNo	string	1	1		body_value1
Dept	string	1	1		body_value2
EmpName	string	1	1		body_value3
Address	string	1	1		body_value4
Tel	string	1	1		body_value5
Salary	string	1	1		body_value6

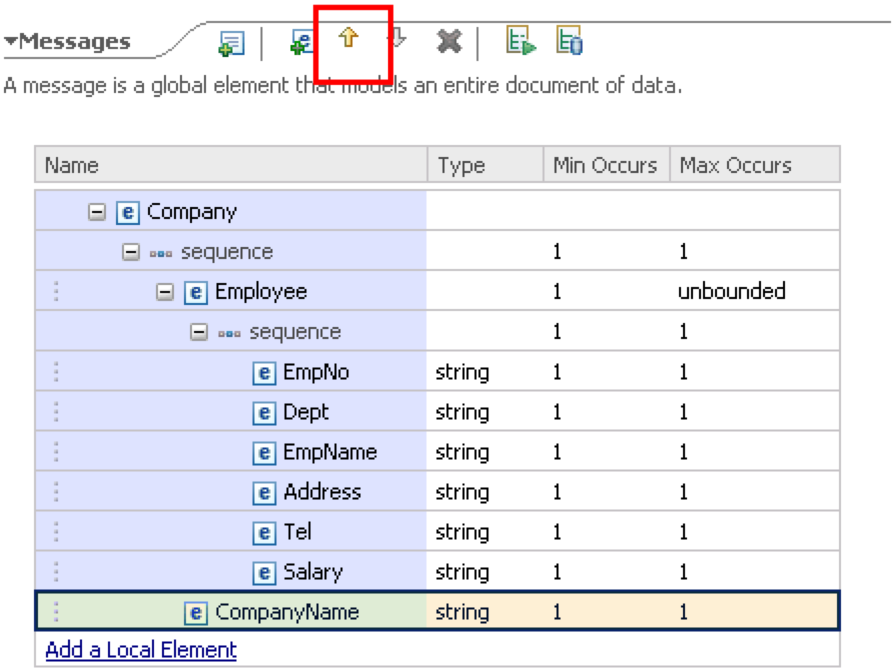
5. Name the new element "CompanyName"



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

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

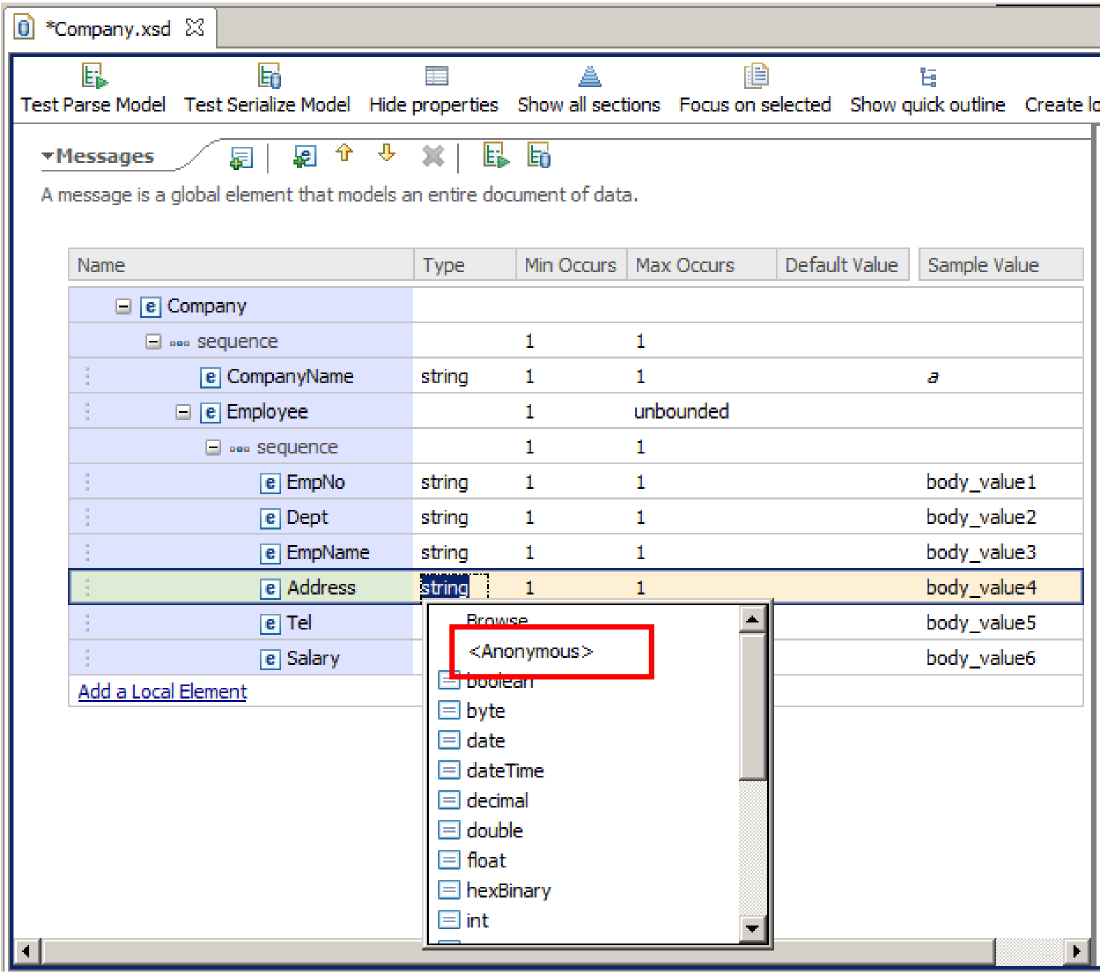
6. Highlight "CompanyName" and click the yellow "Up" arrow to move this element above the "Employee" element (or you can right-click the element and select “Move Up”.)



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

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

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



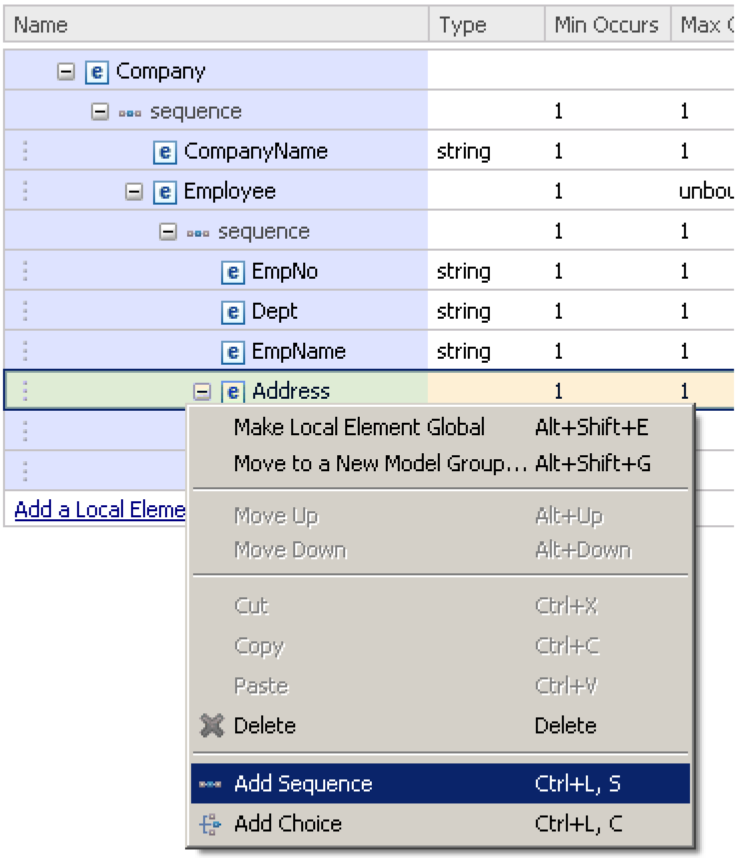
A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
CompanyName	string	1	1		a
Employee		1	unbounded		
sequence		1	1		
EmpNo	string	1	1		body_value1
Dept	string	1	1		body_value2
EmpName	string	1	1		body_value3
Address	string	1	1		body_value4
Tel					body_value5
Salary					body_value6

Add a Local Element

8. Now define three elements under the Address element.

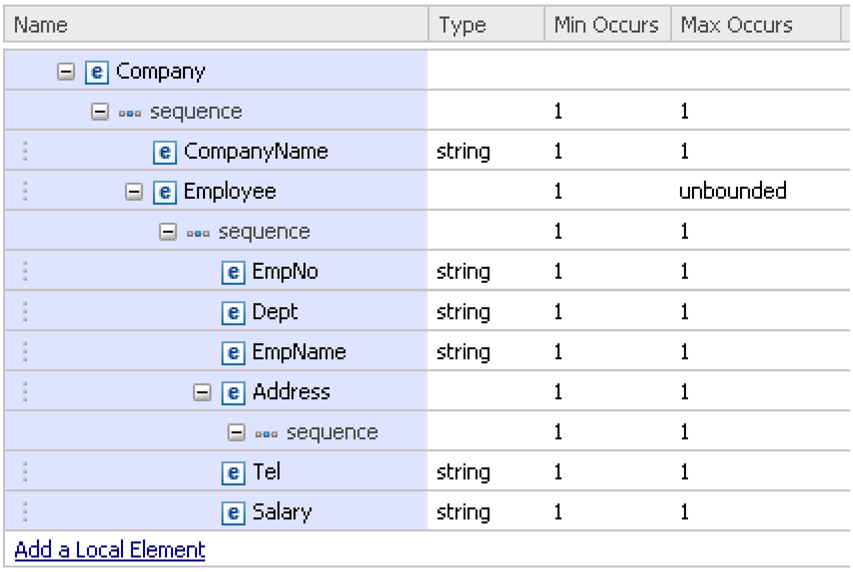
   First, you need to add a new sequence element under Address.
   Right-click on the Address element line (although not on any text) and select "Add Sequence”.



Name	Type	Min Occurs	Max (
⊟ 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 Eleme

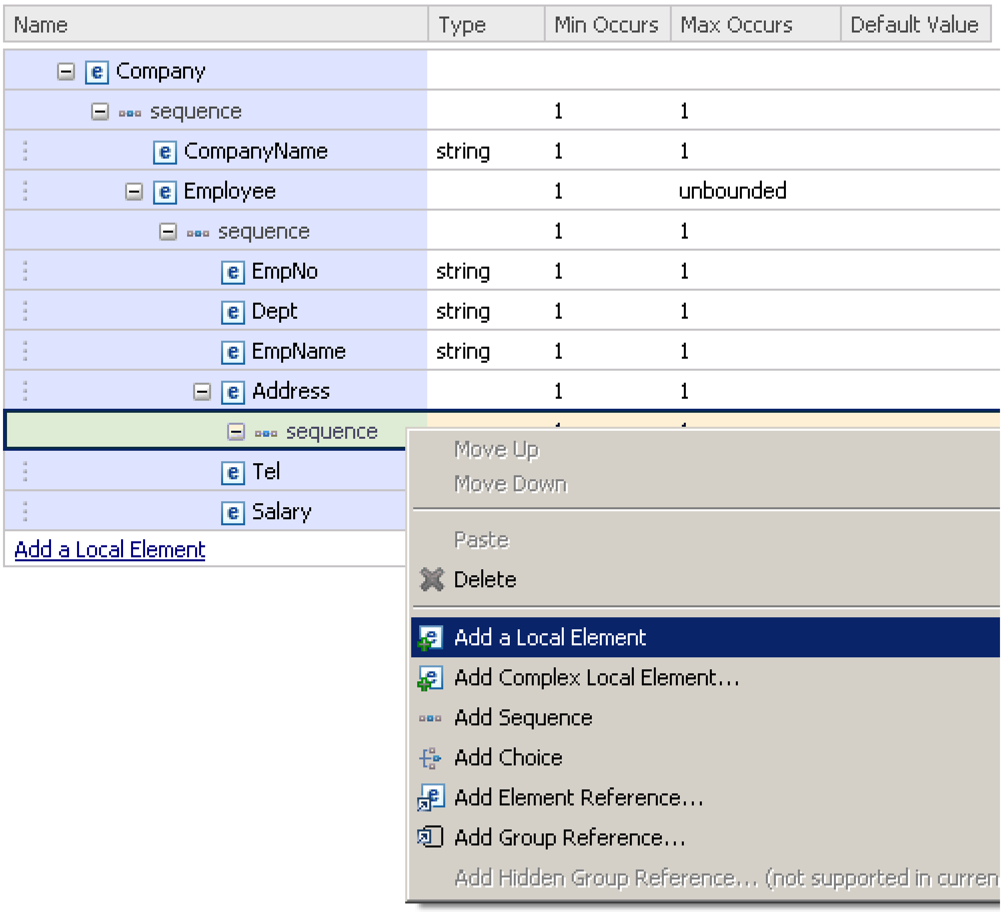
This will be added like this:



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

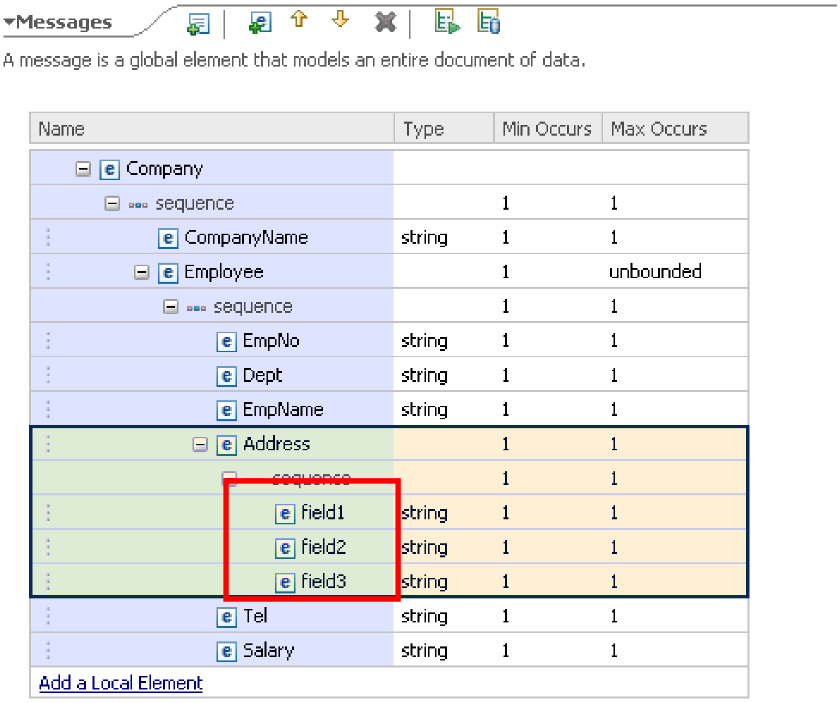
9. Now add a new element under the new Sequence. Right-click on the Sequence line (although not the text *** sequence *** itself), and select Add a Local Element.



Name	Type	Min Occurs	Max Occurs	Default Value
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				
Salary				

Add a Local Element

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



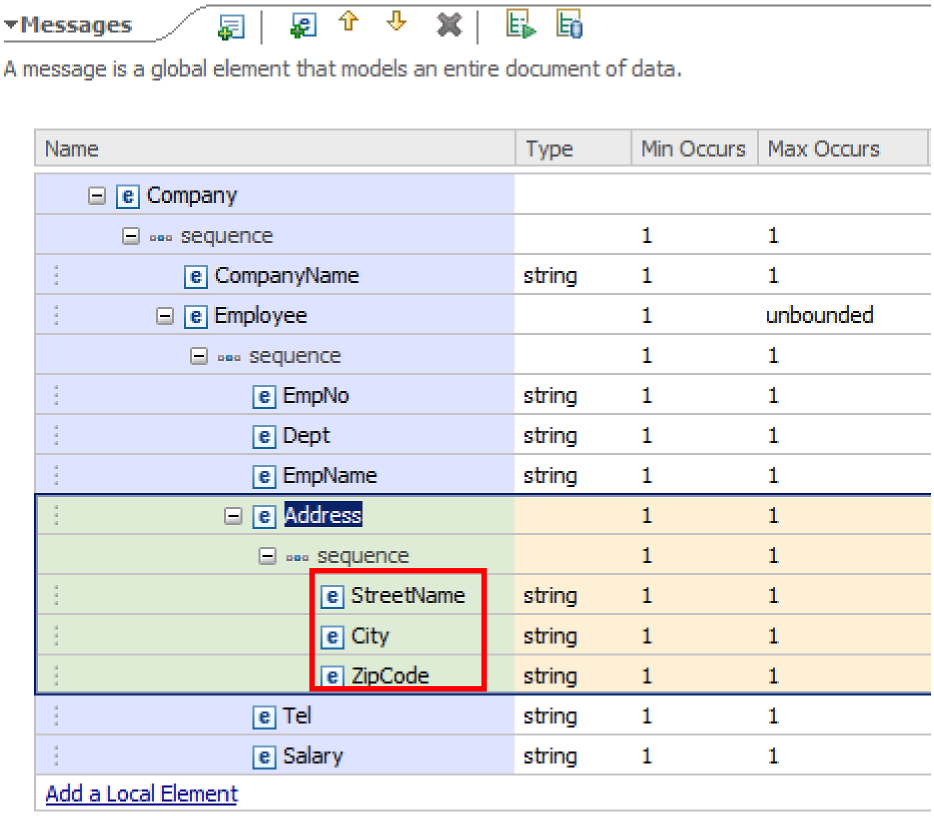
A message is a global element that models an entire document of data.

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
field1	string	1	1
field2	string	1	1
field3	string	1	1
Tel	string	1	1
Salary	string	1	1

Add a Local 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

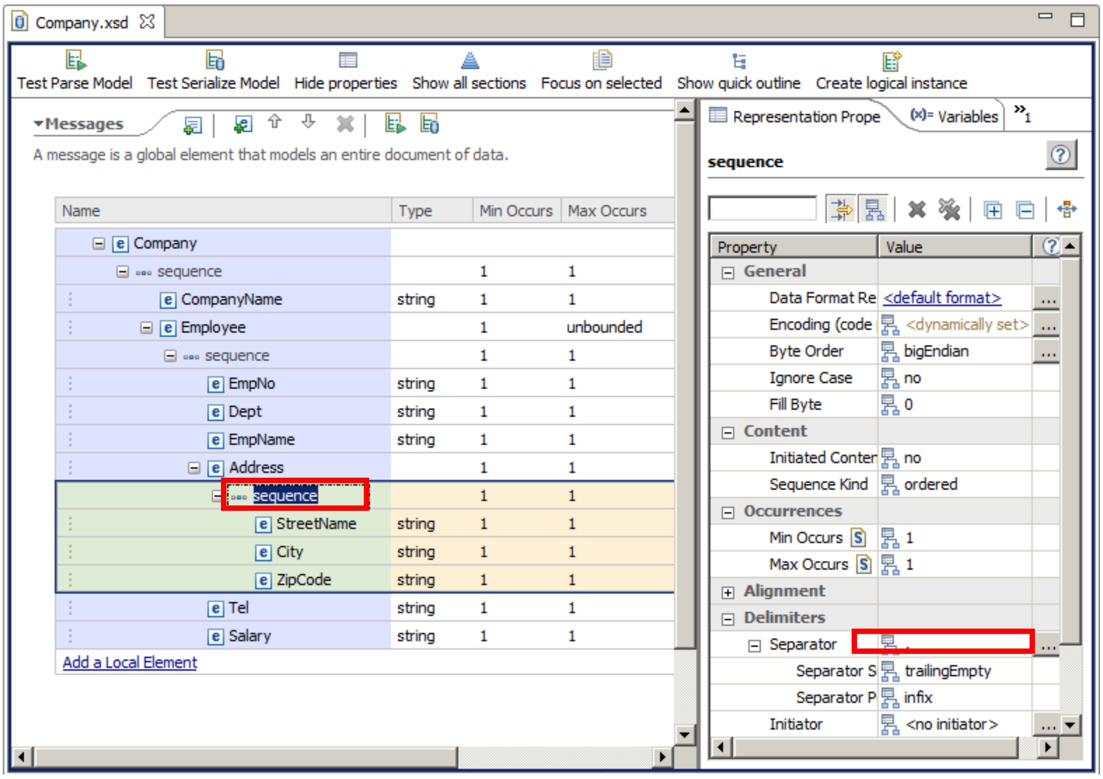


A message is a global element that models an entire document of data.

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
StreetName	string	1	1
City	string	1	1
ZipCode	string	1	1
Tel	string	1	1
Salary	string	1	1

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).

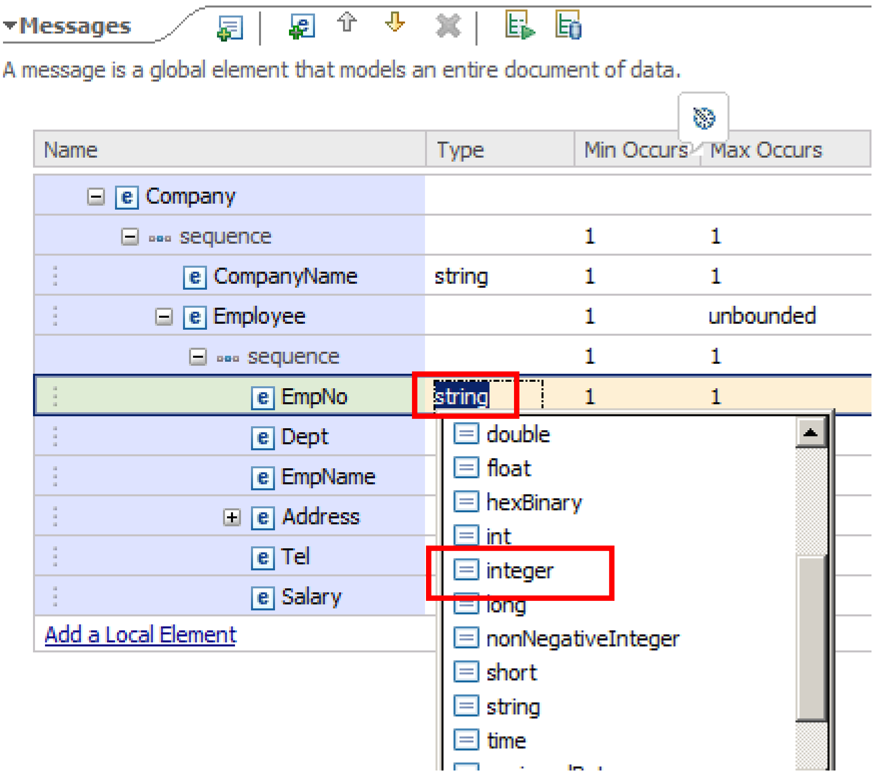


A message is a global element that models an entire document of data.

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
StreetName	string	1	1
City	string	1	1
ZipCode	string	1	1
Tel	string	1	1
Salary	string	1	1

Property	Value
General	
Data Format Re	<default format>
Encoding (code	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	
Initiated Conter	no
Sequence Kind	ordered
Occurrences	
Min Occurs	1
Max Occurs	1
Alignment	
Delimiters	
Separator	,
Separator S	trailingEmpty
Separator P	infix
Initiator	<no initiator>

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



Messages

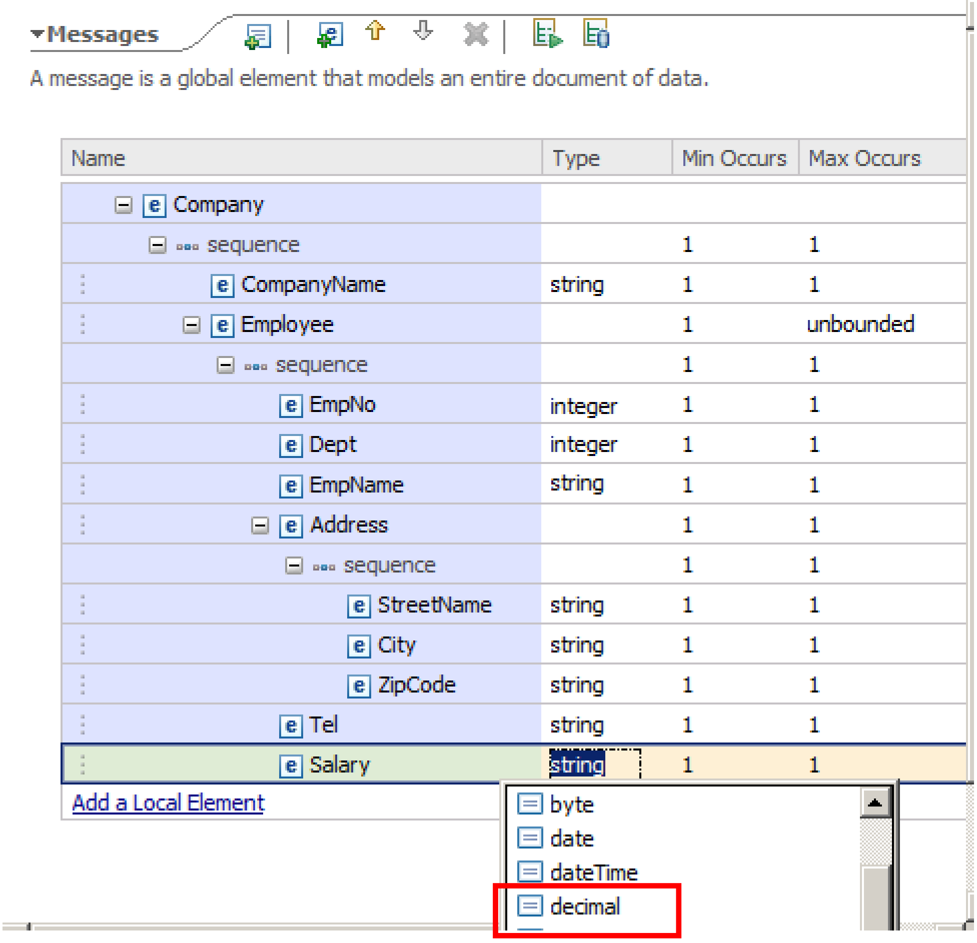
A message is a global element that models an entire document of data.

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			
EmpName			
Address			
Tel			
Salary			

Add a Local Element

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.



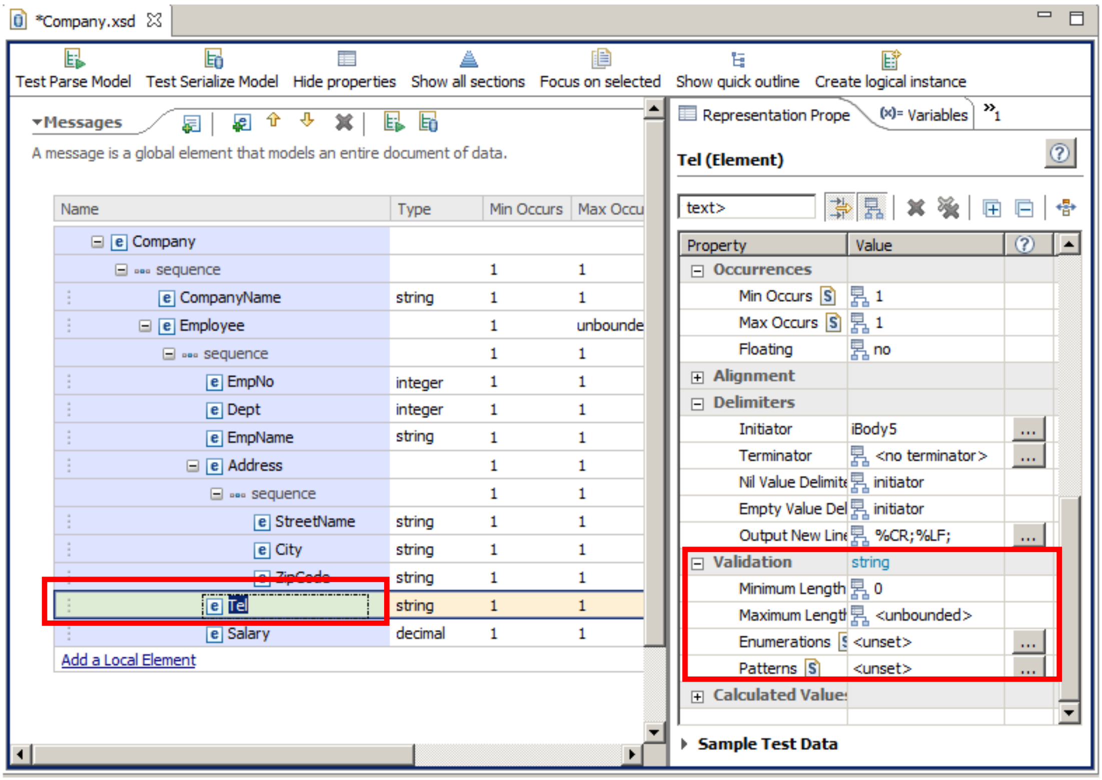
Messages

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

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	string	1	1

Add a Local Element

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



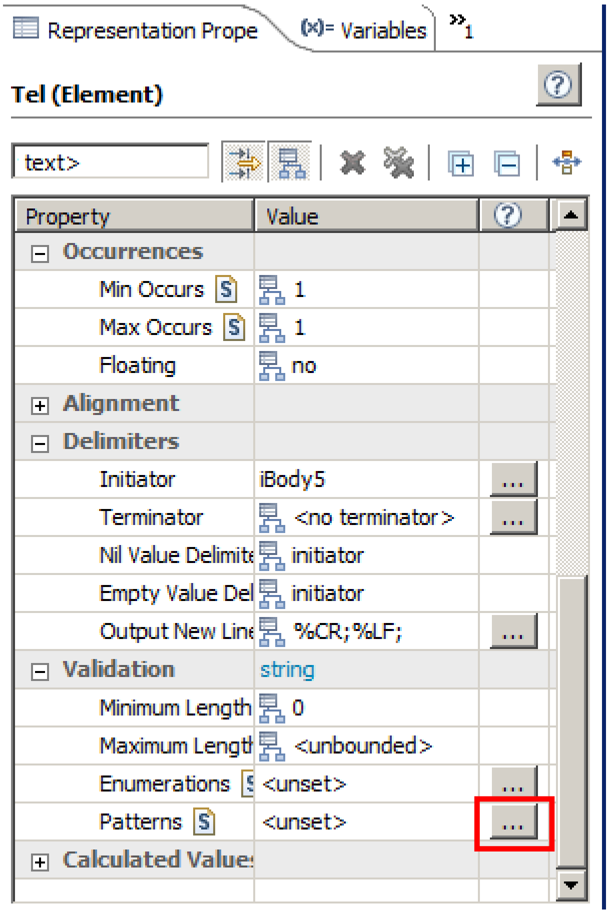
A message is a global element that models an entire document of data.

Name	Type	Min Occurs	Max Occu
Company			
sequence		1	1
CompanyName	string	1	1
Employee		1	unbounde
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

Tel (Element)

Property	Value
Occurrences	
Min Occurs	1
Max Occurs	1
Floating	no
Alignment	
Delimiters	
Initiator	iBody5
Terminator	<no terminator>
Nil Value Delimite	initiator
Empty Value Del	initiator
Output New Line	%CR;%LF;
Validation	string
Minimum Length	0
Maximum Lengt	<unbounded>
Enumerations	<unset>
Patterns	<unset>
Calculated Value:	

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

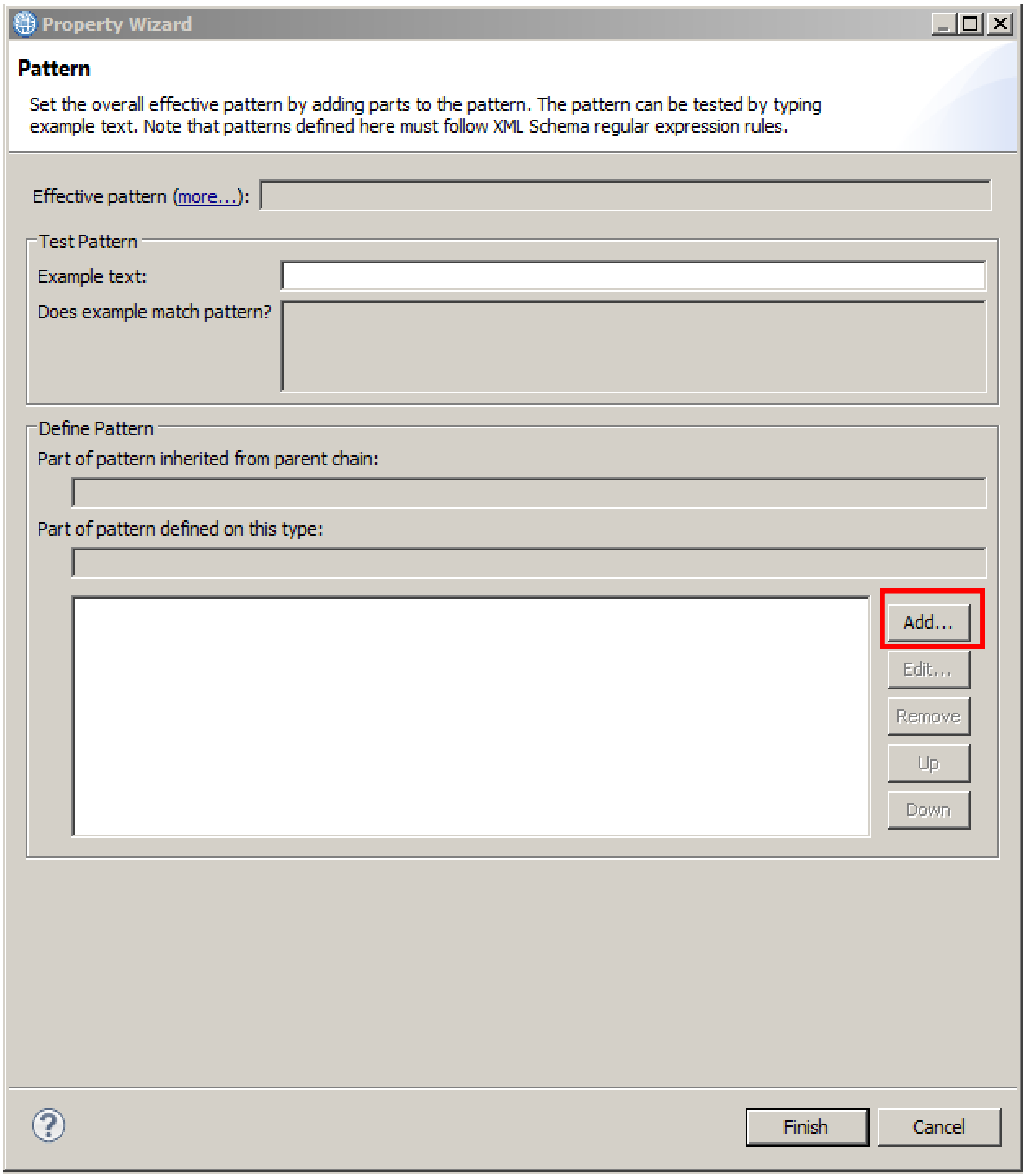


Tel (Element)

text>

Property	Value	
⊟ Occurrences		
Min Occurs	1	
Max Occurs	1	
Floating	no	
⊞ Alignment		
⊟ Delimiters		
Initiator	iBody5	...
Terminator	<no terminator>	...
Nil Value Delimite	initiator	
Empty Value Del	initiator	
Output New Line	%CR;%LF;	...
⊟ Validation	string	
Minimum Length	0	
Maximum Lengtl	<unbounded>	
Enumerations	<unset>	...
Patterns	<unset>	...
⊞ Calculated Value:		

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



Property Wizard

Pattern

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.

Effective pattern (more...):

Test Pattern

Example text:

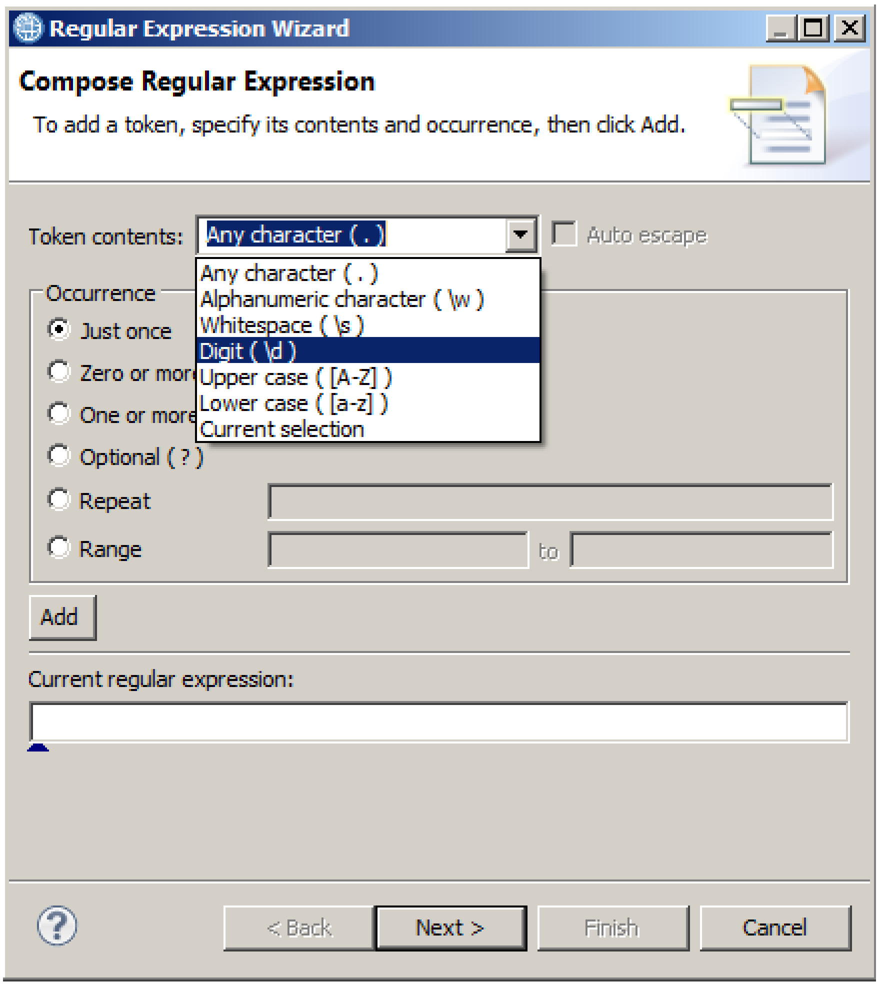
Does example match pattern?

Define Pattern

Part of pattern inherited from parent chain:

Part of pattern defined on this type:

19. 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

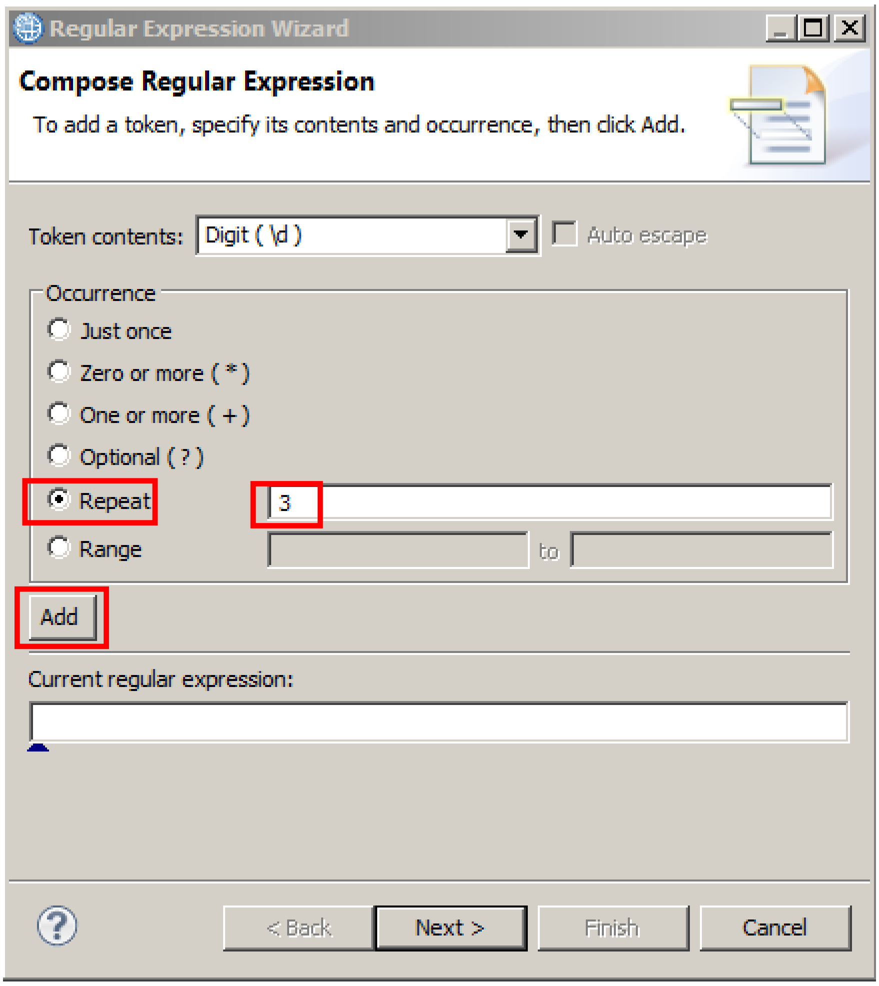
- Any character (.)
- Alphanumeric character (\w)
- Whitespace (\s)
- Digit (\d)
- Upper case ([A-Z])
- Lower case ([a-z])
- Current selection

Occurrence

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

Current regular expression:

20. 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: Digit (\d)

□ Auto escape

Occurrence

○ Just once

○ Zero or more (*)

○ One or more (+)

○ Optional (?)

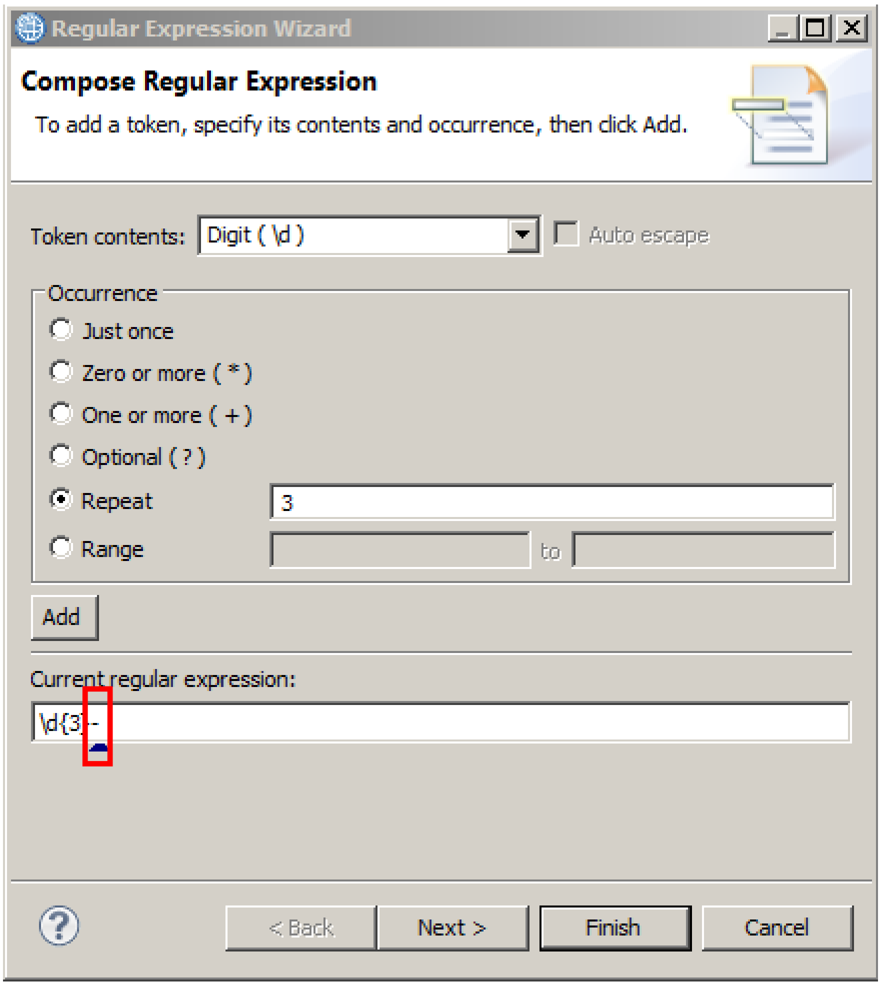
● Repeat 3

○ Range to

Add

Current regular expression:

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



Regular Expression Wizard

Compose Regular Expression

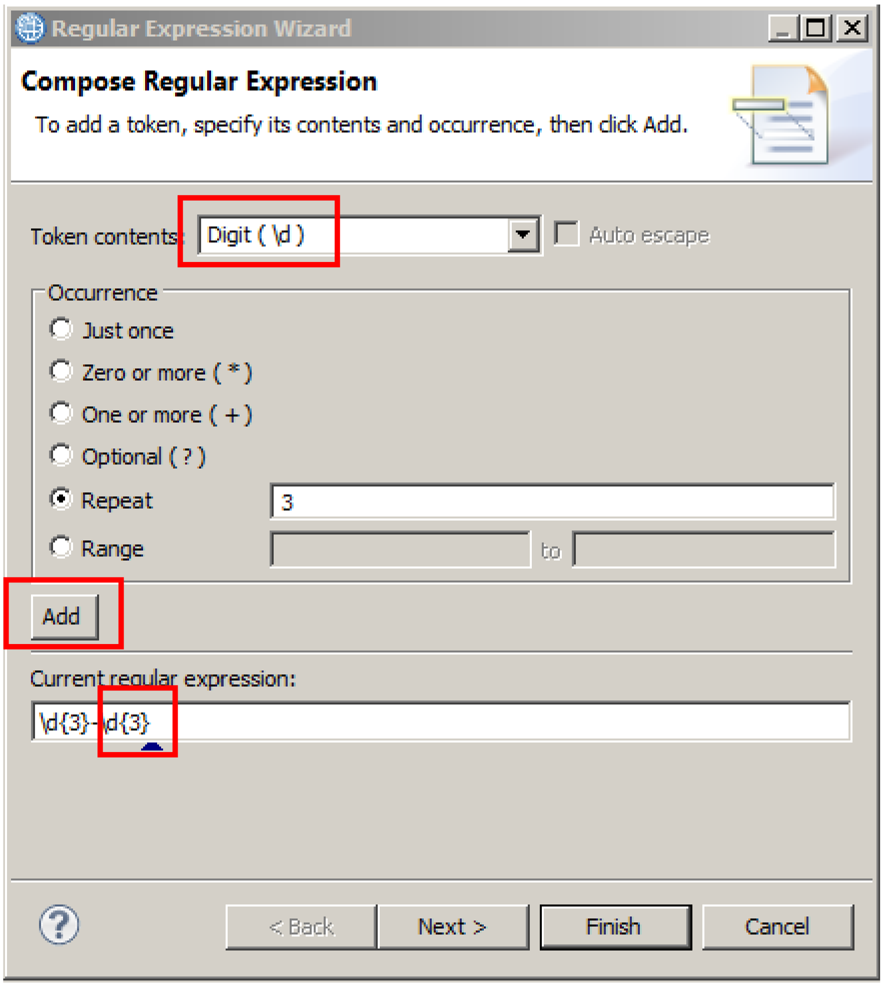
To add a token, specify its contents and occurrence, then click Add.

Token contents: Digit (\d) □ Auto escape

Occurrence
- ○ Just once
- ○ Zero or more (*)
- ○ One or more (+)
- ○ Optional (?)
- ● Repeat 3
- ○ Range to

Current regular expression: \d{3}-

22. 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: Digit (\d) □ Auto escape

Occurrence

○ Just once

○ Zero or more (*)

○ One or more (+)

○ Optional (?)

● Repeat 3

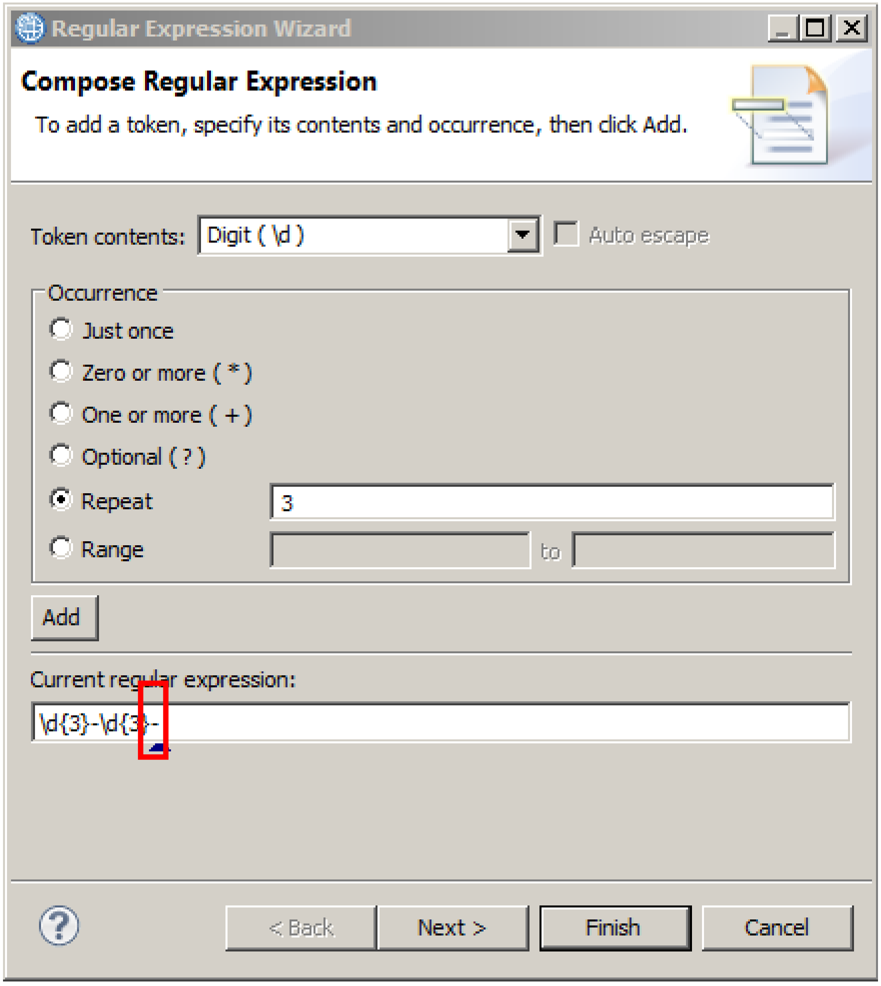
○ Range to

Add

Current regular expression:

\d{3}-\d{3}

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

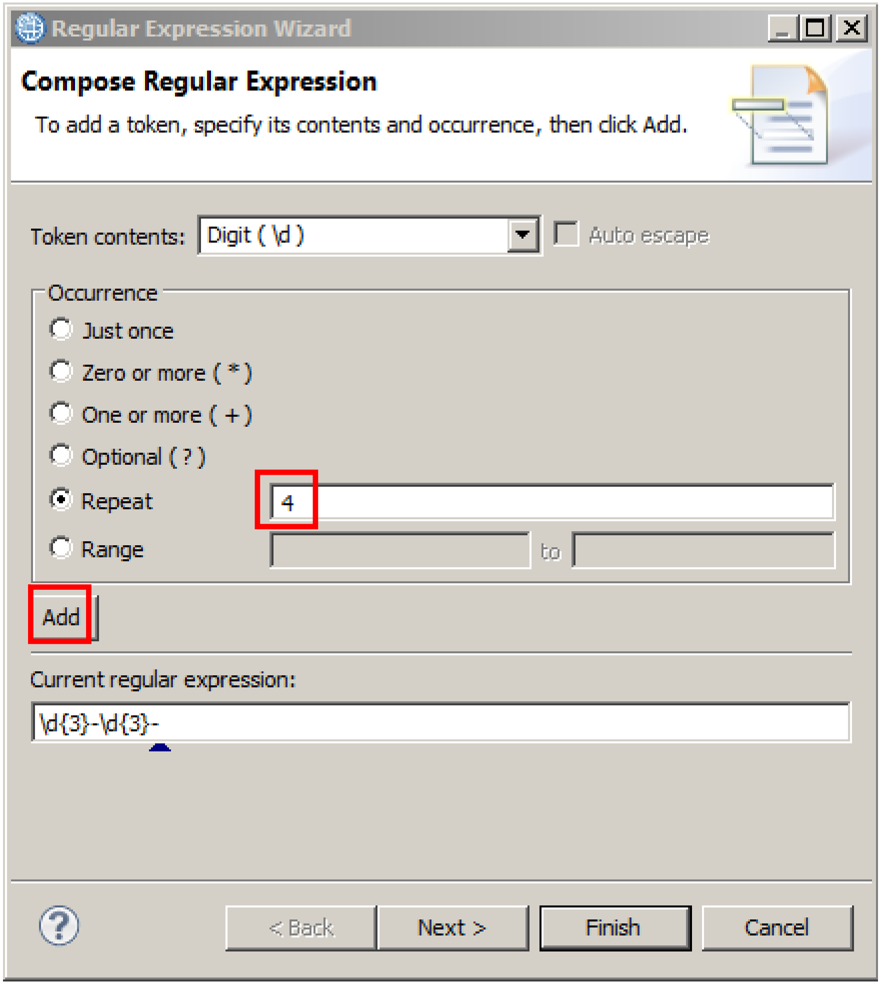


Token contents: Digit (\d)

Repeat: 3

Current regular expression: \d{3}-\d{3}-

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.



Compose Regular Expression

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

Token contents: Digit (\d) ☐ Auto escape

Occurrence

○ Just once

○ Zero or more (*)

○ One or more (+)

○ Optional (?)

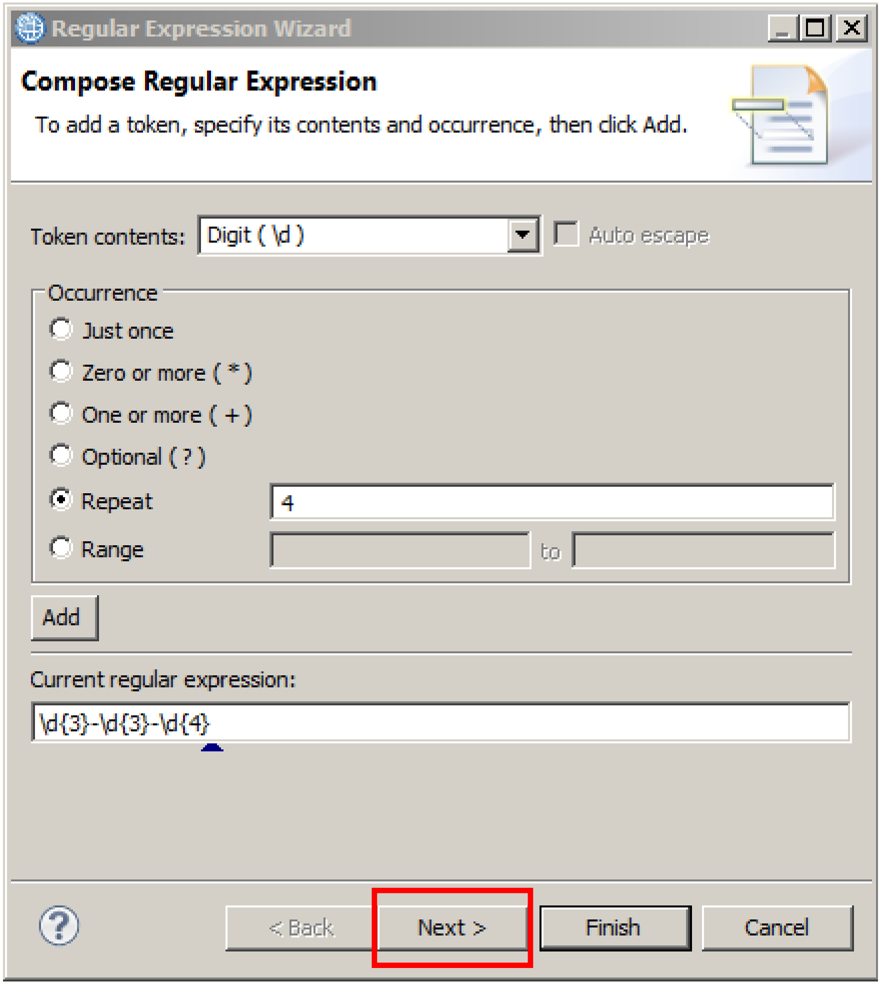
● Repeat 4

○ Range to

Current regular expression:

\d{3}-\d{3}-

25. Click on the Next button.



Compose Regular Expression

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

Token contents: Digit (\d) □ Auto escape

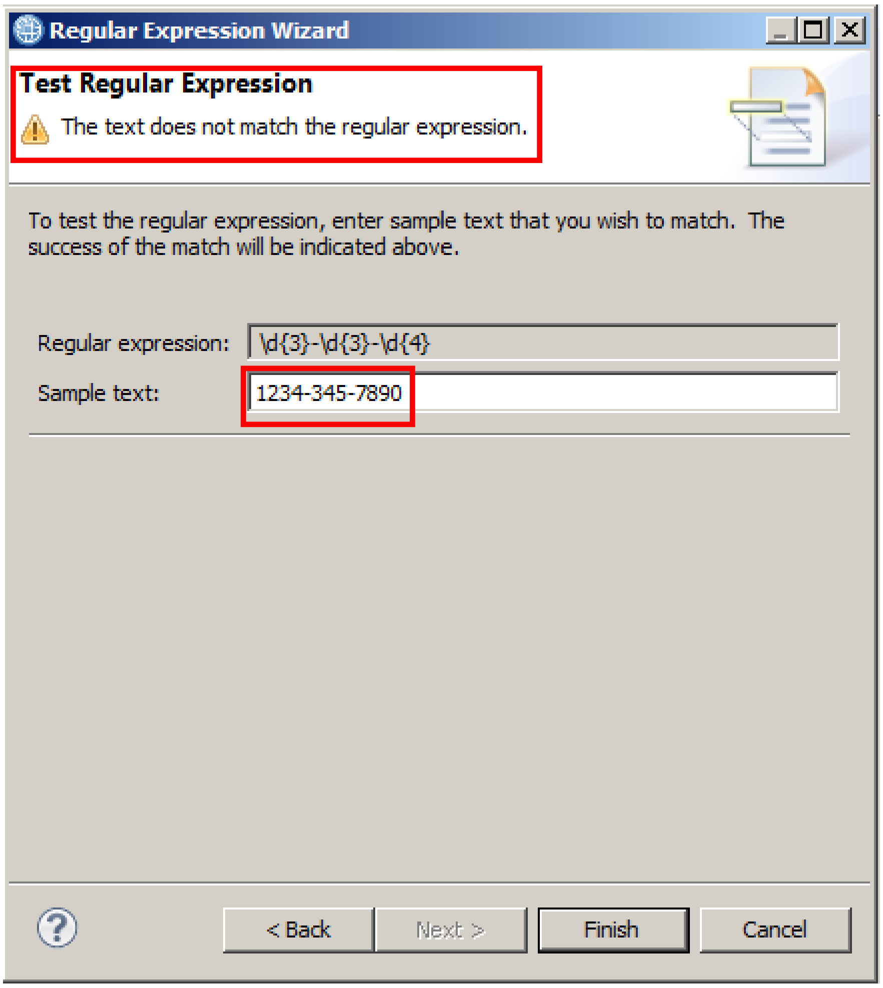
Occurrence
○ Just once
○ Zero or more (*)
○ One or more (+)
○ Optional (?)
● Repeat 4
○ Range to

Current regular expression:

```
\d{3}-\d{3}-\d{4}
```

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".



Regular Expression Wizard

Test Regular Expression

⚠ The text does not match the regular expression.

To test the regular expression, enter sample text that you wish to match. The success of the match will be indicated above.

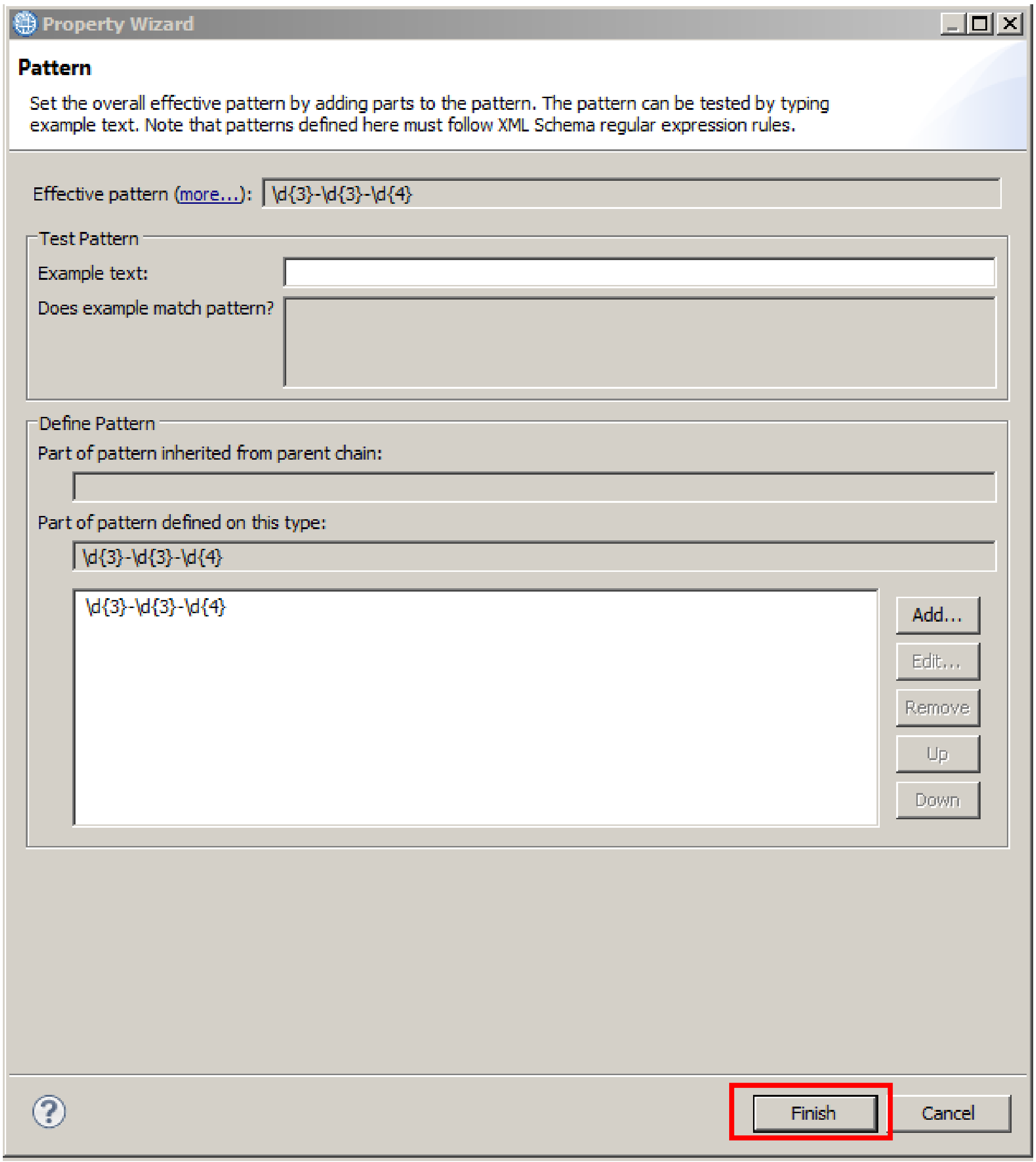
Regular expression: \d{3}-\d{3}-\d{4}

Sample text: 1234-345-7890

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.



Property Wizard

Pattern

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.

Effective pattern (more...): \d{3}-\d{3}-\d{4}

Test Pattern

Example text:

Does example match pattern?

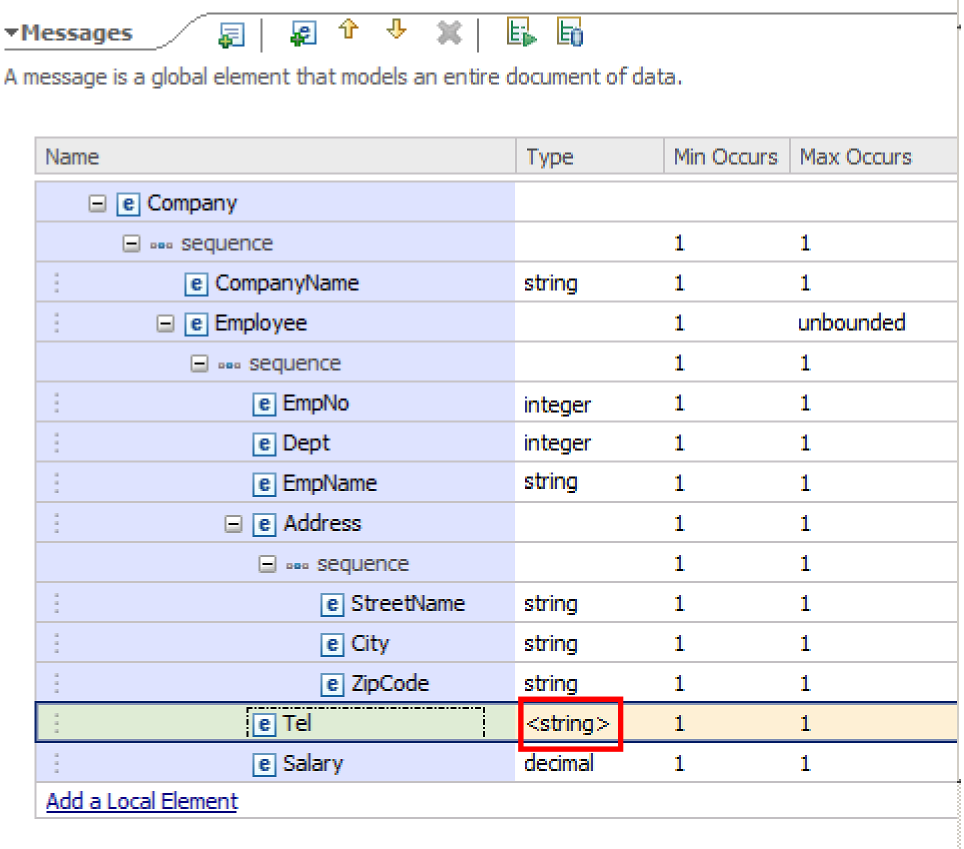
Define Pattern

Part of pattern inherited from parent chain:

Part of pattern defined on this type: \d{3}-\d{3}-\d{4}

\d{3}-\d{3}-\d{4}

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.

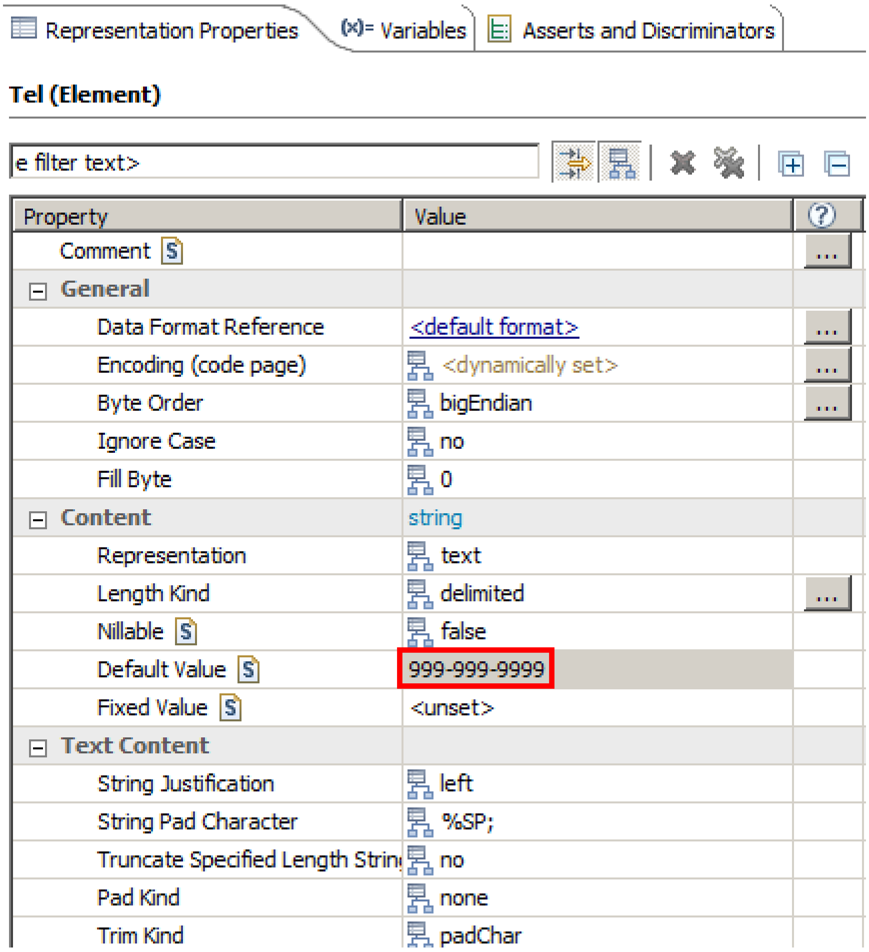


A message is a global element that models an entire document of data.

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

Add a Local Element

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).



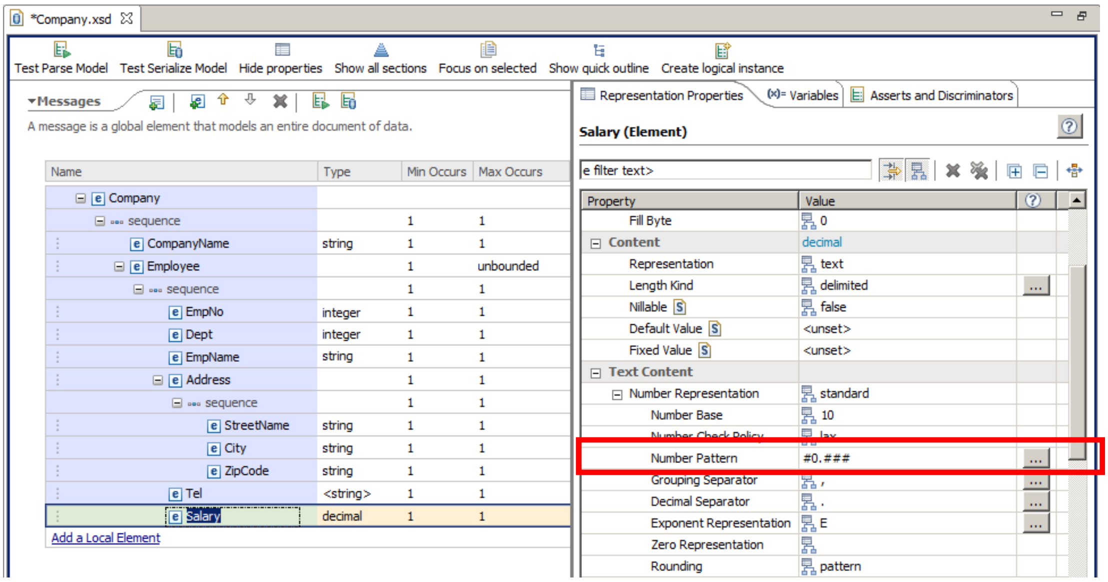
Tel (Element)

Property	Value
Comment	
General	
Data Format Reference	<default format>
Encoding (code page)	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	string
Representation	text
Length Kind	delimited
Nillable	false
Default Value	999-999-9999
Fixed Value	<unset>
Text Content	
String Justification	left
String Pad Character	%SP;
Truncate Specified Length Strin	no
Pad Kind	none
Trim Kind	padChar

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".



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

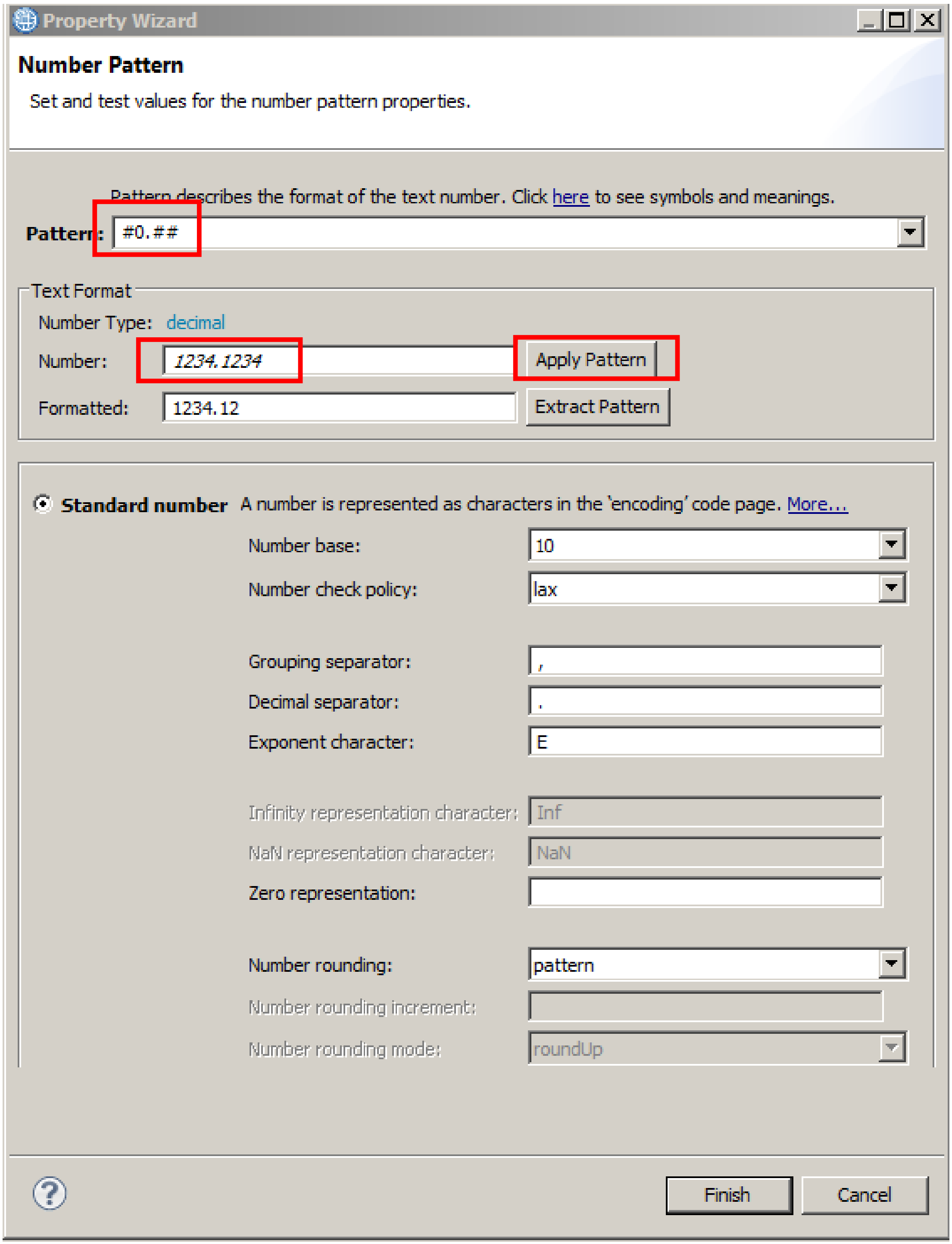
Name	Type	Min Occurs	Max Occurs
Company			
sequence		1	1
CompanyName	string	1	1
Employee		1	unbounded
sequence		1	1
EmpNo	integer	1	1
Dept	integer	1	1
EmpName	string	1	1
Address		1	1
sequence		1	1
StreetName	string	1	1
City	string	1	1
ZipCode	string	1	1
Tel	<string>	1	1
Salary	decimal	1	1

Salary (Element)

Property	Value
Fill Byte	0
Content	decimal
Representation	text
Length Kind	delimited
Nillable	false
Default Value	<unset>
Fixed Value	<unset>
Text Content	
Number Representation	standard
Number Base	10
Number Check Policy	lax
Number Pattern	#0.###
Grouping Separator	,
Decimal Separator	.
Exponent Representation	E
Zero Representation	
Rounding	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.



Number Pattern

Set and test values for the number pattern properties.

Pattern describes the format of the text number. Click here to see symbols and meanings.

Pattern: #0.##

Text Format

Number Type: decimal

Number: 1234.1234

Formatted: 1234.12

● **Standard number** A number is represented as characters in the 'encoding' code page. More...

Number base: 10

Number check policy: lax

Grouping separator: ,

Decimal separator: .

Exponent character: E

Infinity representation character: Inf

NaN representation character: NaN

Zero representation:

Number rounding: pattern

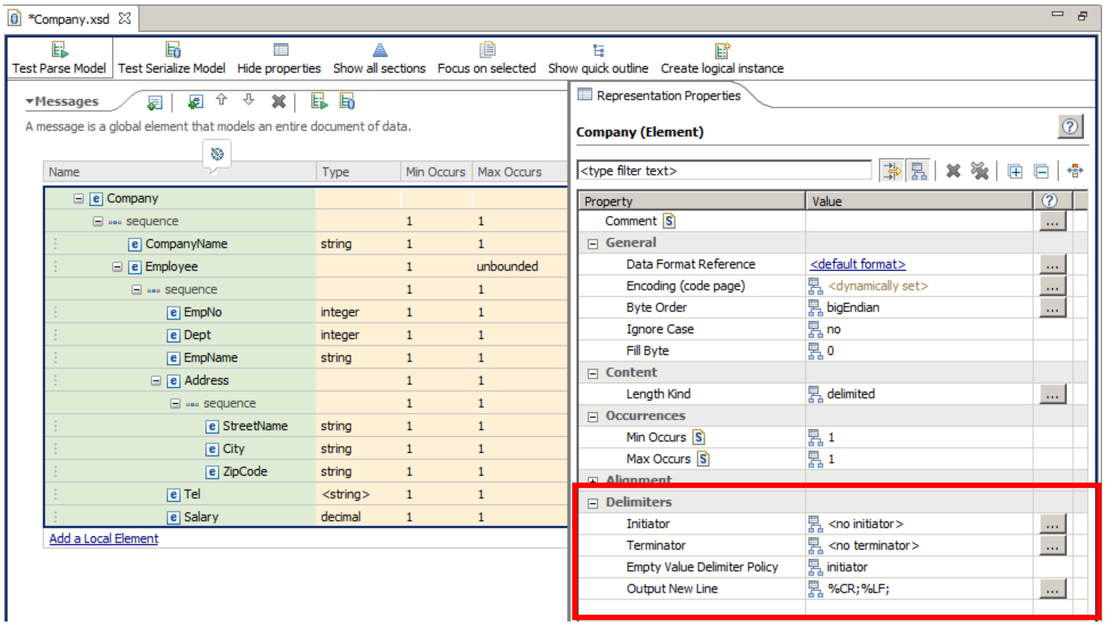
Number rounding increment:

Number rounding mode: roundUp

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.



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

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

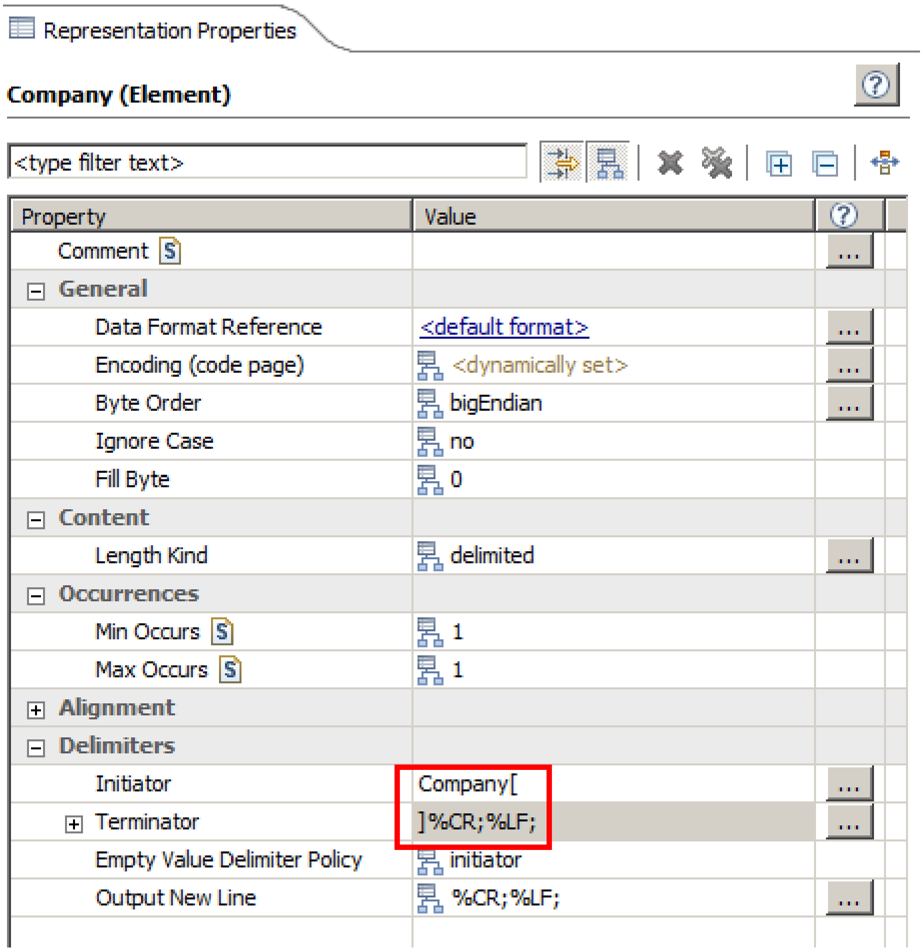
Company (Element)

Property	Value
Comment	
General	
Data Format Reference	<default format>
Encoding (code page)	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	
Length Kind	delimited
Occurrences	
Min Occurs	1
Max Occurs	1
Alignment	
Delimiters	
Initiator	<no initiator>
Terminator	<no terminator>
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;

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.



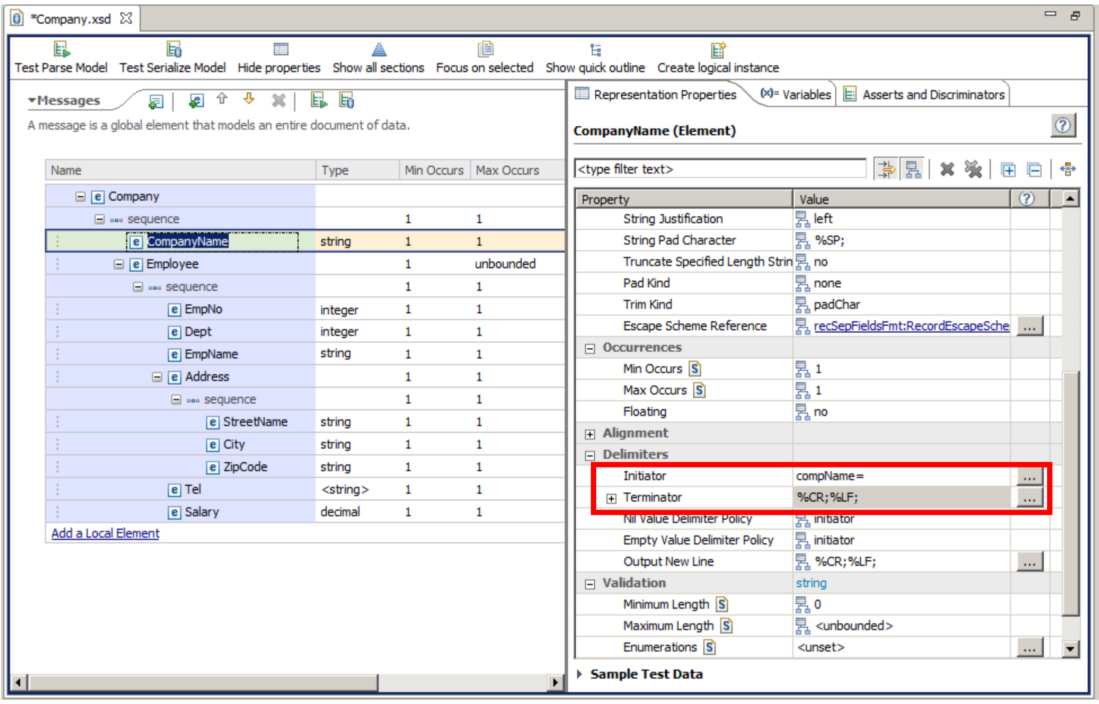
Representation Properties

Company (Element)

<type filter text>

Property	Value	?
Comment		...
⊟ **General**		
Data Format Reference	<default format>	...
Encoding (code page)	<dynamically set>	...
Byte Order	bigEndian	...
Ignore Case	no	
Fill Byte	0	
⊟ **Content**		
Length Kind	delimited	...
⊟ **Occurrences**		
Min Occurs	1	
Max Occurs	1	
⊞ **Alignment**		
⊟ **Delimiters**		
Initiator	Company[	...
⊞ Terminator	]%CR;%LF;	...
Empty Value Delimiter Policy	initiator	
Output New Line	%CR;%LF;	...

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:



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

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

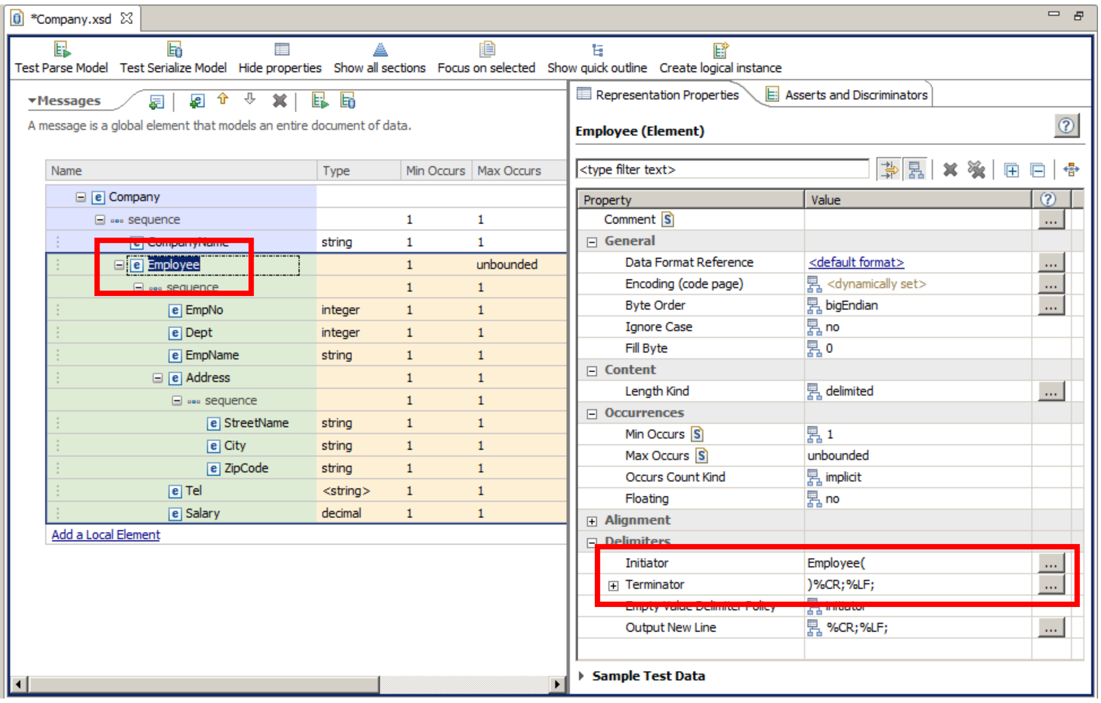
CompanyName (Element)

<type filter text>

Property	Value
String Justification	left
String Pad Character	%SP;
Truncate Specified Length Strin	no
Pad Kind	none
Trim Kind	padChar
Escape Scheme Reference	recSepFieldsFmt:RecordEscapeSche
Occurrences	
Min Occurs	1
Max Occurs	1
Floating	no
Alignment	
Delimiters	
Initiator	compName=
Terminator	%CR;%LF;
Nil Value Delimiter Policy	initiator
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;
Validation	string
Minimum Length	0
Maximum Length	<unbounded>
Enumerations	<unset>

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.



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

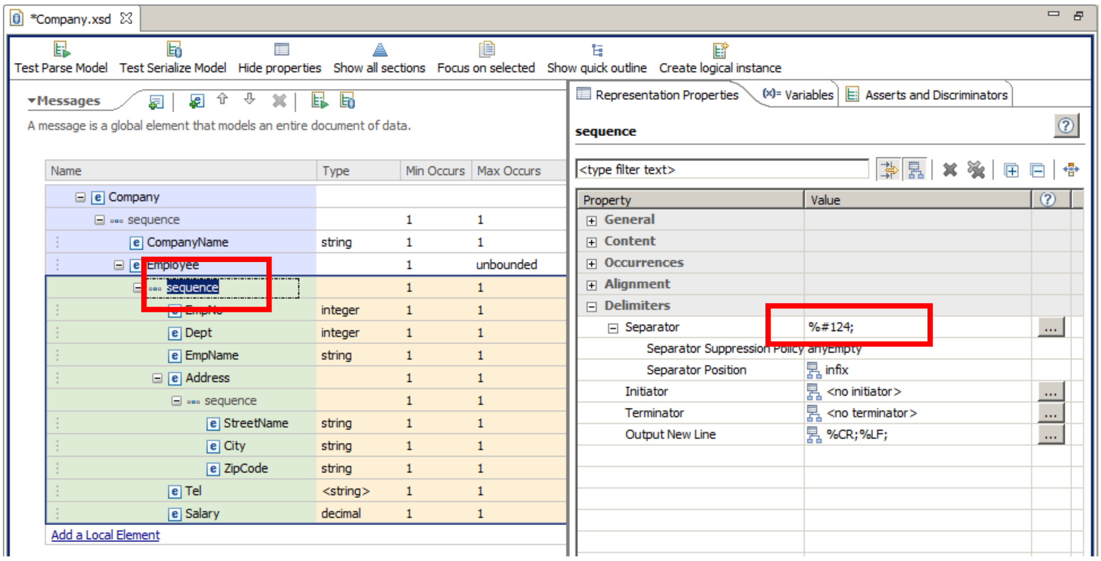
Name	Type	Min Occurs	Max Occurs
Company			
sequence		1	1
CompanyName	string	1	1
Employee		1	unbounded
sequence		1	1
EmpNo	integer	1	1
Dept	integer	1	1
EmpName	string	1	1
Address		1	1
sequence		1	1
StreetName	string	1	1
City	string	1	1
ZipCode	string	1	1
Tel	<string>	1	1
Salary	decimal	1	1

Employee (Element)

<type filter text>

Property	Value
Comment	
General	
Data Format Reference	<default format>
Encoding (code page)	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	
Length Kind	delimited
Occurrences	
Min Occurs	1
Max Occurs	unbounded
Occurs Count Kind	implicit
Floating	no
Alignment	
Delimiters	
Initiator	Employee(
Terminator	)%CR;%LF;
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;

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).



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

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

Add a Local Element

sequence

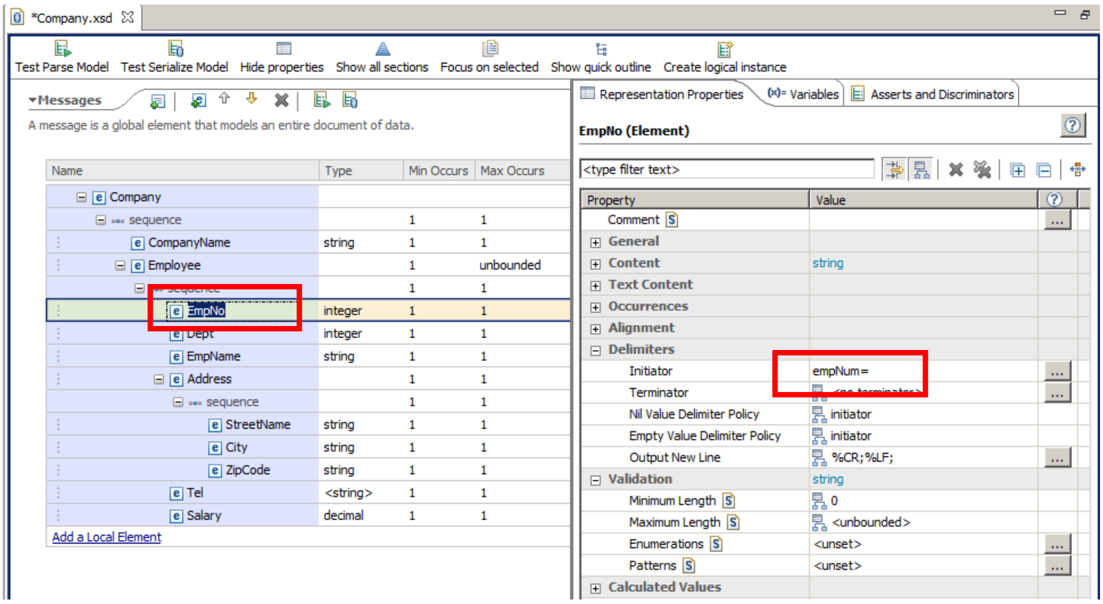
Property	Value
General	
Content	
Occurrences	
Alignment	
Delimiters	
Separator	%#124;
Separator Suppression Policy	anyEmpty
Separator Position	infix
Initiator	<no initiator>
Terminator	<no terminator>
Output New Line	%CR;%LF;

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”.



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

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

Add a Local Element

EmpNo (Element)

Property	Value
Comment	
General	
Content	string
Text Content	
Occurrences	
Alignment	
Delimiters	
Initiator	empNum=
Terminator	<no terminator>
Nil Value Delimiter Policy	initiator
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;
Validation	string
Minimum Length	0
Maximum Length	<unbounded>
Enumerations	<unset>
Patterns	<unset>
Calculated Values	

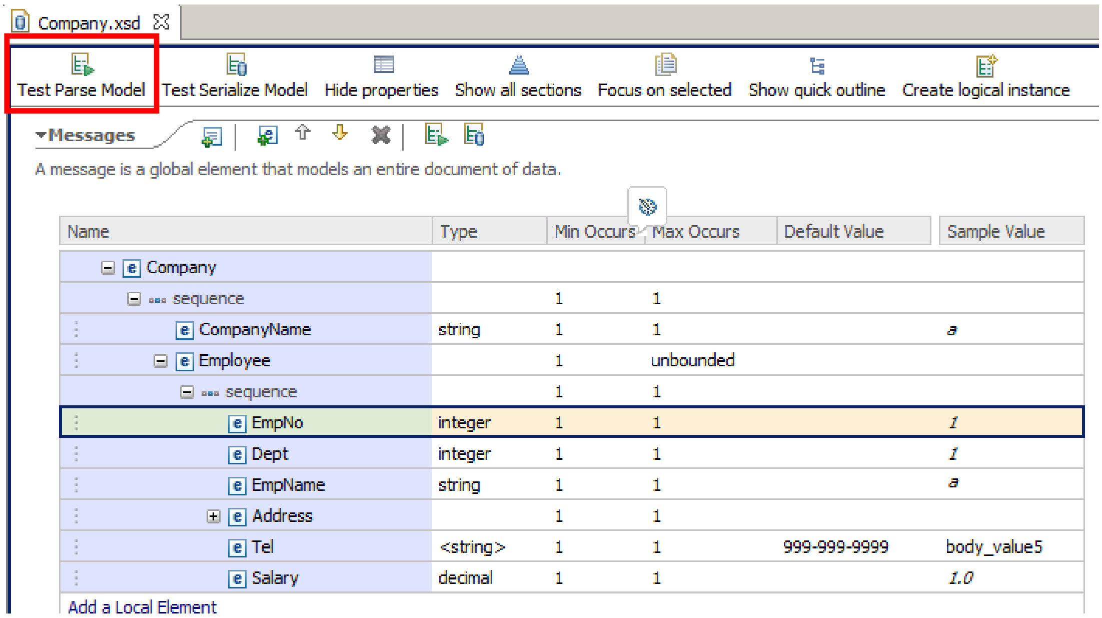
38. Save your DFDL Schema by pressing Ctrl+S or File->Save. When saved, the DFDL Schema is validated and if any errors (or warnings) are found, they will appear in the Problems view.

    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.

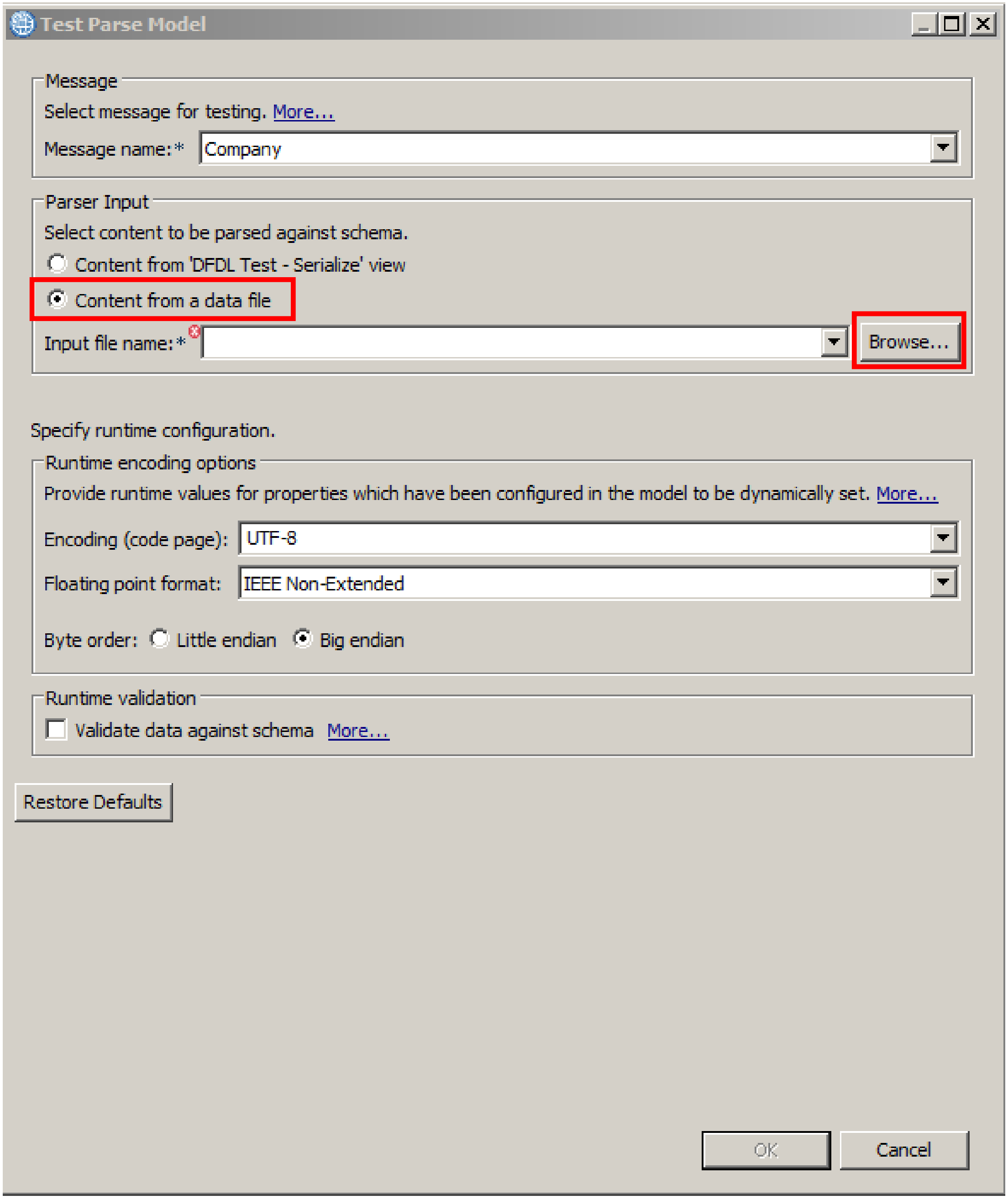
Click on the "Test Parse Model".



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

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
CompanyName	string	1	1		a
Employee		1	unbounded		
sequence		1	1		
EmpNo	integer	1	1		1
Dept	integer	1	1		1
EmpName	string	1	1		a
Address		1	1		
Tel	<string>	1	1	999-999-9999	body_value5
Salary	decimal	1	1		1.0

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:*

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

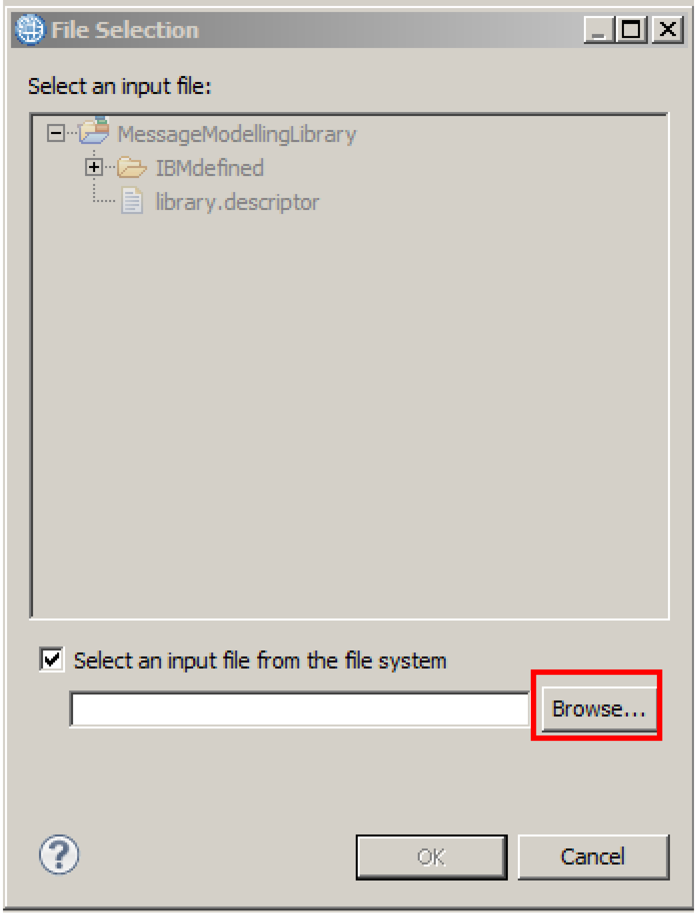
Floating point format: IEEE Non-Extended

Byte order: ○ Little endian ● Big endian

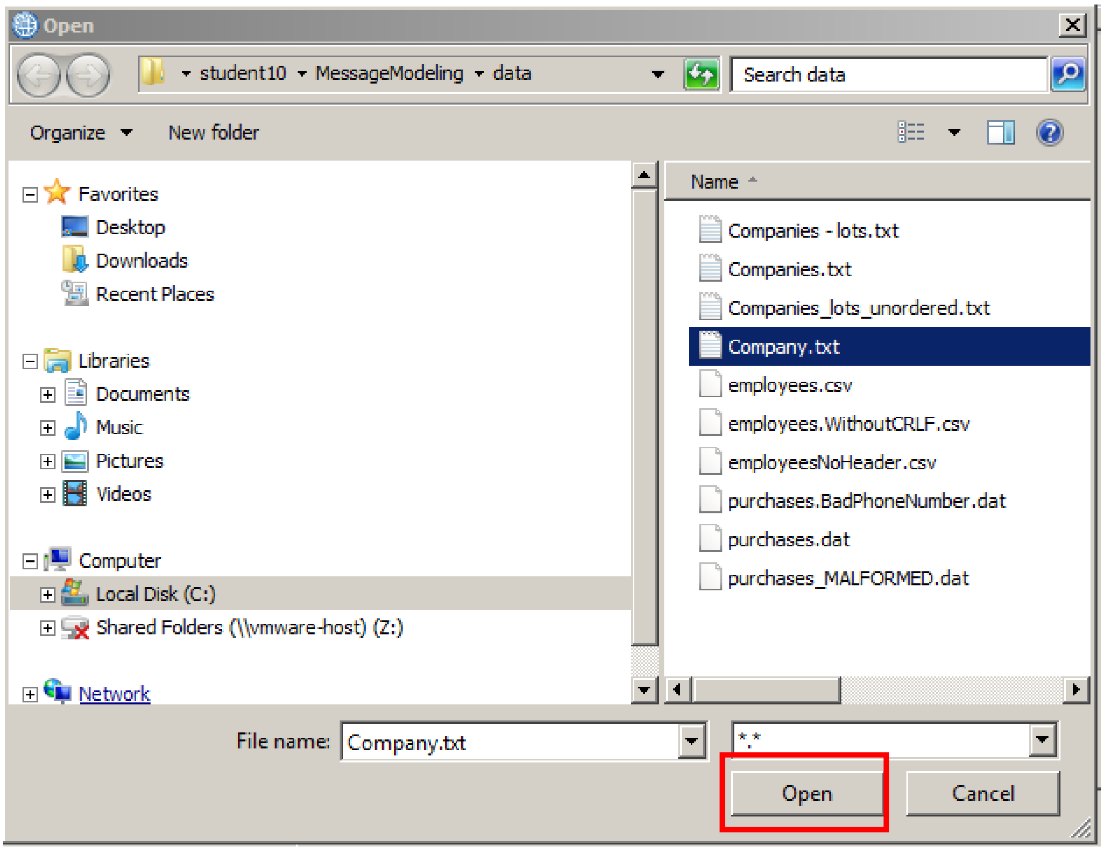
Runtime validation

□ Validate data against schema More...

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.

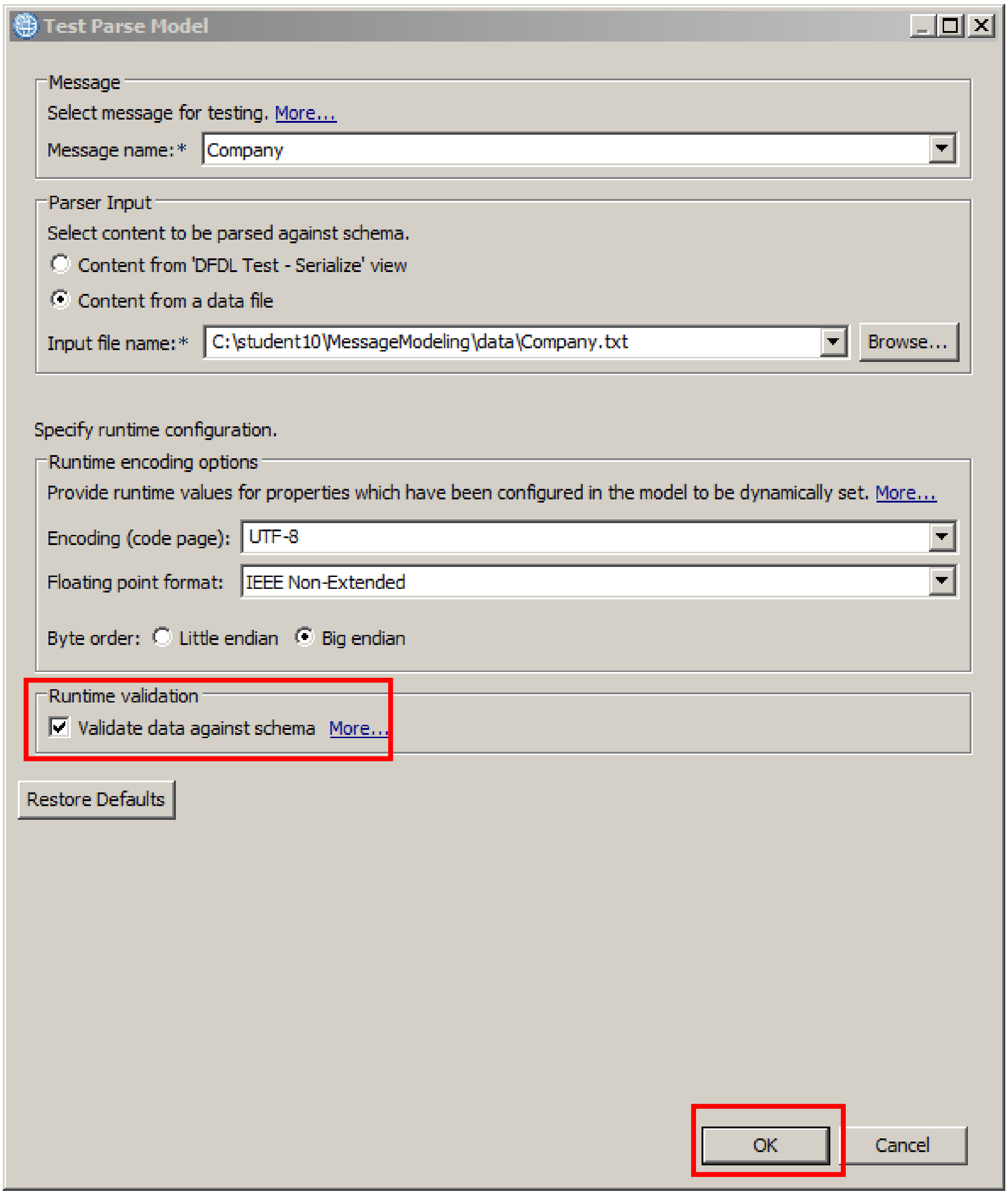


File name: Company.txt

.

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

   Click OK.



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:\student10\MessageModeling\data\Company.txt

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

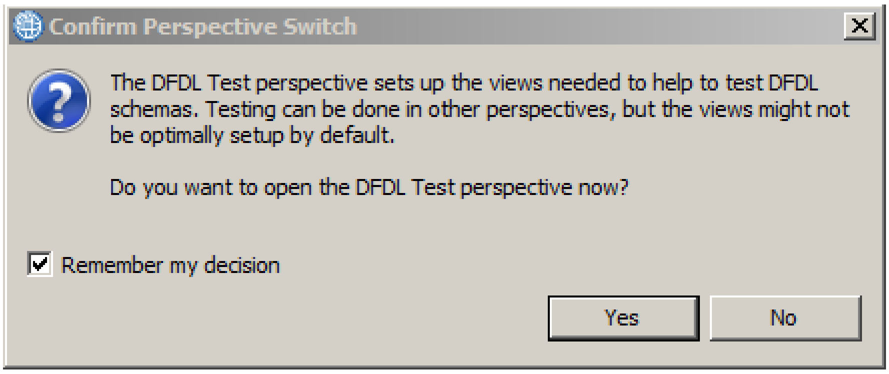
Floating point format: IEEE Non-Extended

Byte order: ○ Little endian ● Big endian

Runtime validation

☑ Validate data against schema More...

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



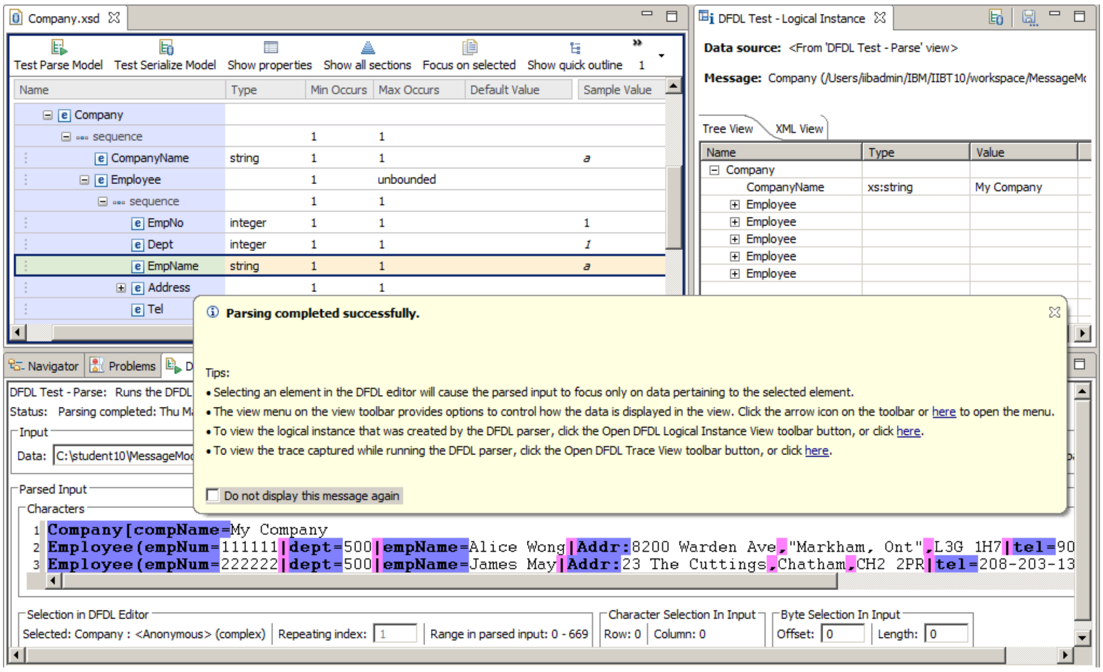
Confirm Perspective Switch

The DFDL Test perspective sets up the views needed to help to test DFDL schemas. Testing can be done in other perspectives, but the views might not be optimally setup by default.

Do you want to open the DFDL Test perspective now?

☑ Remember my decision

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



Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Company					
sequence		1	1		
CompanyName	string	1	1		a
Employee		1	unbounded		
sequence		1	1		
EmpNo	integer	1	1		1
Dept	integer	1	1		1
EmpName	string	1	1		a
Address		1	1		
Tel					

Data source: <From 'DFDL Test - Parse' view>

Message: Company (/Users/iibadmin/IBM/IIBT10/workspace/MessageMo

Name	Type	Value
Company		
CompanyName	xs:string	My Company
Employee		
Employee		
Employee		
Employee		
Employee		

Parsing completed successfully.

Tips:

- Selecting an element in the DFDL editor will cause the parsed input to focus only on data pertaining to the selected element.
- The view menu on the view toolbar provides options to control how the data is displayed in the view. Click the arrow icon on the toolbar or here to open the menu.
- To view the logical instance that was created by the DFDL parser, click the Open DFDL Logical Instance View toolbar button, or click here.
- To view the trace captured while running the DFDL parser, click the Open DFDL Trace View toolbar button, or click here.

□ Do not display this message again

DFDL Test - Parse: Runs the DFDL

Status: Parsing completed: Thu M

Input

Data: C:\student10\MessageMod

Parsed Input

Characters

```
Company[compName=My Company
Employee(empNum=111111|dept=500|empName=Alice Wong|Addr:8200 Warden Ave,"Markham, Ont",L3G 1H7|tel=90
Employee(empNum=222222|dept=500|empName=James May|Addr:23 The Cuttings,Chatham,CH2 2PR|tel=208-203-13
```

Selection in DFDL Editor

Selected: Company : <Anonymous> (complex) | Repeating index: 1 | Range in parsed input: 0 - 669

Character Selection In Input

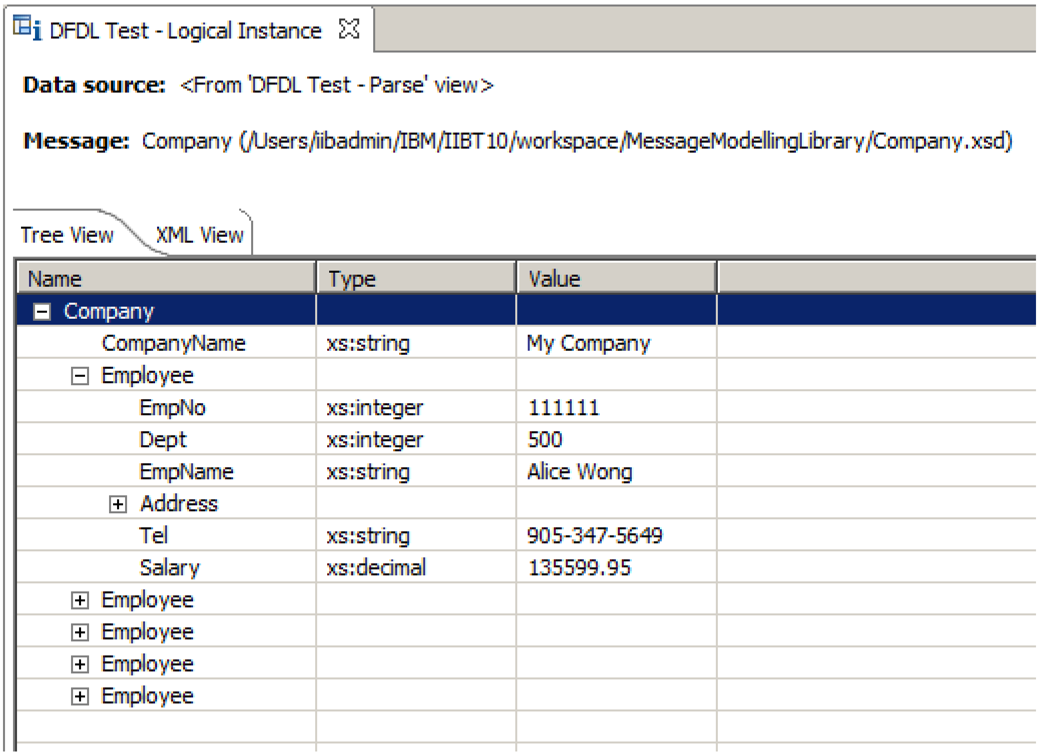
Row: 0 | Column: 0

Byte Selection In Input

Offset: 0 | Length: 0

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.



Data source: <From 'DFDL Test - Parse' view>

Message: Company (/Users/iibadmin/IBM/IIBT10/workspace/MessageModellingLibrary/Company.xsd)

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