

IBM Integration Bus

Message Modeling with DFDL

Lab 4

Record-oriented, tagged, delimited text (advanced)

Featuring:

- Schema references
- Initiators, separators, terminators
- Arrays - element count provided explicitly within data
- Unordered sequences
- Arrays - element count given by parsing data

June 2015

Hands-on lab built at product
Version 10.0.0.0

1. INTRODUCTION	3
1.1 LAB OBJECTIVES	3
1.2 LAB PREPARATION.....	3
1.3 TDS BASIC LAB RECAP	4
1.4 TDS ADVANCED SCENARIO	5
2. CREATE THE MESSAGE MODEL IN THE LIBRARY	6
3. REFINE THE MESSAGE MODEL	12
4. TESTING THE MESSAGE MODEL	31
5. ESTABLISHING REPEATING ELEMENTS BY PARSING	37
6. UNORDERED SEQUENCES	42
END OF LAB GUIDE	45

1. Introduction

1.1 Lab objectives

In this lab you are going to extend the message model you created in the previous (basic TDS) lab, Lab3.

The lab examines more complex structures, such as schema references, element delimiters like initiators, separators and terminators, and the ability to handle variable numbers of elements.

Chapters 5 and 6 contains scenarios that are supported only in IIB V9.0.0.2 and later (inc V10). These scenarios are arrays where the count does not occur in the data and is established solely by parsing (by setting the Occurs Count Kind property to "parsed"), and sequences where the elements occur in any order (by setting the Sequence Kind property to "unordered").

1.2 Lab preparation

If you did not do Lab 3 (the basic TDS lab), you can import a pre-built solution. Import the PI file

```
c:\student10\message_modelling\solution\  
MessageModelSolution_TDS_Basic_Lab3(SharedLib).zip
```

If you wish to proceed straight to the new function provided in IIB V9002 (Occurs Count Kind=parsed in chapter 5 and unordered sequences in chapter 6), import the partial solution of this lab:

```
c:\student10\message_modelling\solution\  
MessageModelSolution_TDS_Advanced_Lab4(SharedLib-PartialSolution).zip
```

A complete solution of this lab is provided in:

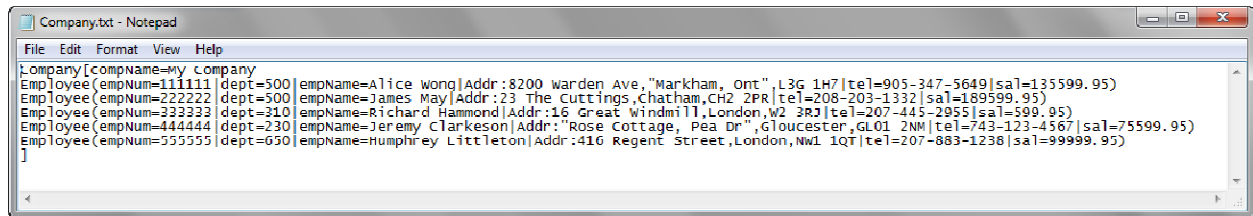
```
c:\student10\message_modelling\solution\  
MessageModelSolution_TDS_Advanced_Lab4(SharedLib-FullSolution).zip
```

If you import one of the PI files provided above, the imported library will be called

```
MessageModelling_TDS
```

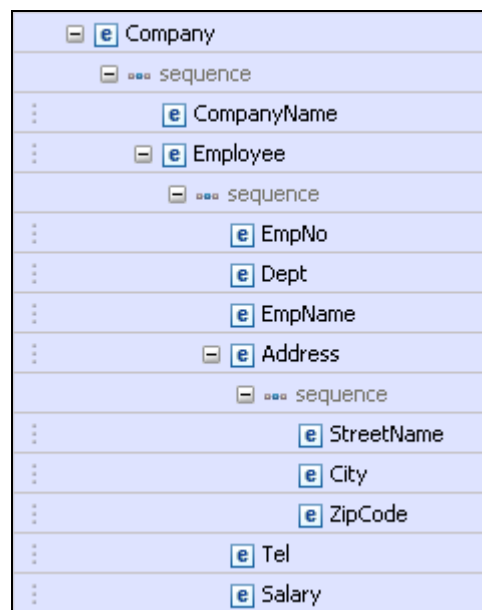
1.3 TDS Basic Lab Recap

In the TDS basic lab, you created a message model to parse this data file. The file has a single Company record which contains multiple Employee records.



```
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=300|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 3R2|tel=207-443-2955|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)
]
```

So you defined the following DFDL structure like this:



1.4 TDS Advanced Scenario

In this lab, you will extend the message model created in the Basic scenario so that it can parse the following data file, which contains multiple Company records.

In the first part of this lab (chapter 3), the number of Company records is specified by the compCount element in the Header record (the supplied test file has a value of 5).

```

Companies.txt - Notepad
File Edit Format View Help
Header{recDesc:My Company records,compCount:5}
Company[compName=BBC
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=6189599.95)
Employee(empNum=333333|dept=310|empName=Richard Hamurid|Addr:16 Great Windmill St,Lynduri,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=5599.95)
Employee(empNum=555555|dept=650|empName=Humphrey Littleton|Addr:416 Regent Street,London,NW1 1QT|tel=207-883-1238|sal=99999.95)
]
Company[compName=IBM
Employee(empNum=111111|dept=9876|empName=Arnold Buzby|Addr:1000 The Close,Winchester,L3G 1H7|tel=905-345-5649|sal=23.54)
Employee(empNum=222222|dept=2350|empName=Digby Jones|Addr:1 Porstmouth Rd,Southampton,CH2 2PR|tel=208-203-1332|sal=599.95)
]
Company[compName=Big Bank
Employee(empNum=111111|dept=1|empName=Mr Big|Addr:99 Sicillian Dr,"Palermo, NY",L3G 1H7|tel=905-347-5649|sal=4599.00)
Employee(empNum=000001|dept=1|empName=Homer Simpson|Addr:Two's complement,Springfield,1011001|tel=208-203-1332|sal=189599.94)
Employee(empNum=333333|dept=2|empName=Lucy Wetherell|Addr:23 Gas St,Bolton,W2 3RJ|tel=207-445-2955|sal=599.95)
]
Company[compName=Huge Store
Employee(empNum=111111|dept=20|empName=George Formby|Addr:1 HotPot Rd,Lancashire,L3G 1H7|tel=905-347-5649|sal=85599.95)
Employee(empNum=222222|dept=18|empName=Ivor Engine|Addr:1234 London Rd,LLangollen,CH2 2PR|tel=208-203-1332|sal=9.95)
]
Company[compName=Corner Store
Employee(empNum=100001|dept=4456|empName=Captain Flack|Addr:The Fire Station,Trumpton,L3G 1H7|tel=905-347-5649|sal=12345.95)
Employee(empNum=100002|dept=4429|empName=Captain Pugwash|Addr:The Black Pig,Smugglers Cove,CH2 2PR|tel=208-203-1332|sal=654321.91)
Employee(empNum=100003|dept=4420|empName=Lady Penelope|Addr:creighton ward Mansion,Buckinghamshire,W2 3RJ|tel=207.445.2955|sal=599.95)
Employee(empNum=100004|dept=4483|empName=Jeff Tracy|Addr:Tracy Island,Pacific Ocean,GL012NM|tel=743-123-4567|sal=75599.23)
Employee(empNum=100005|dept=4400|empName=Gordon Tracy|Addr:tracy island,pacific ocean,nw1 1qi|tel=207-883-1238|sal=666.67)
]
Trailer{chksum:1234567890}

```

So you will need to create a structure like this. The Companies schema will use a schema reference to the Company schema that was generated in Lab3.



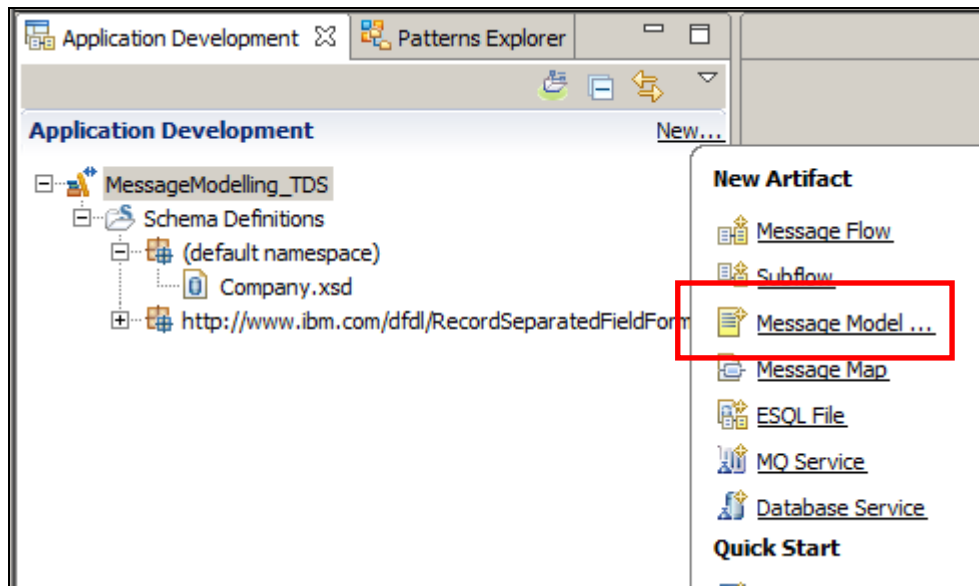
In the next part, chapter 5 shows you how to automatically detect the number of company records. The DFDL parser can be defined to detect the number of Company elements without using the compCount element.

Finally, chapter 6 shows you how to switch the order of the Header elements without having to redefine the order of the elements in the model.

2. Create the Message Model in the Library

1. Use either the library you created for Lab 3 (MessageModelingLibrary), or the provided solution for Lab 3 (MessageModelling_TDS).

Click New->Message Model to create a new DFDL Schema.



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

New Message Model

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

? < Back Next > Finish Cancel

- From the wizard, select "Create a DFDL schema file using the wizard to guide you" and click Next.

New Message Model

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.

Header

CREATED{ DATE=08/22/2011 1:45:23 PM | COUNT=5 }<CRLF>

Body

START{ ID=1 | STATUS=SENT | QTY=300 }<CRLF>

Trailer

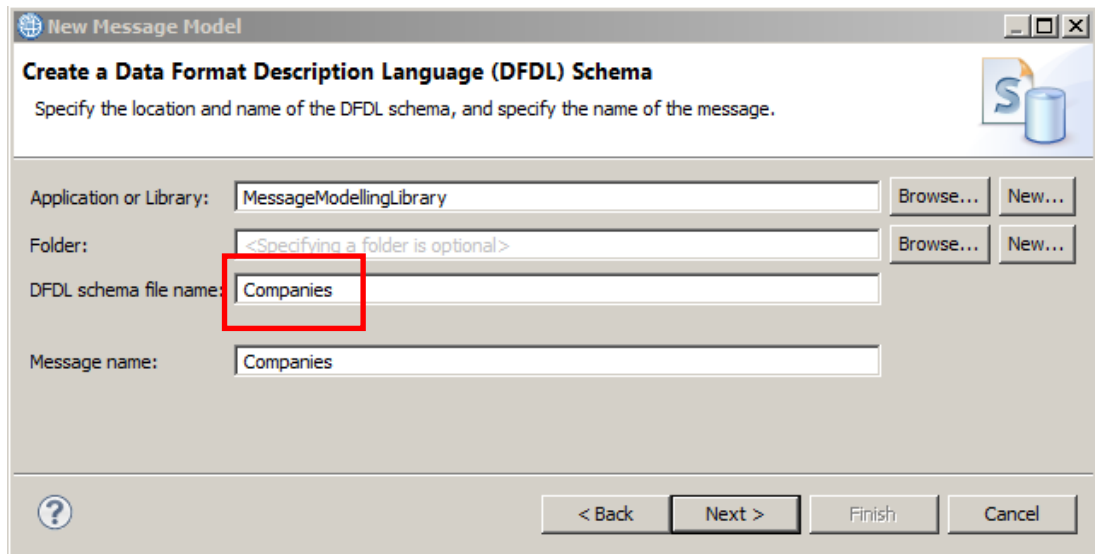
EDITED{ DATE=08/22/2011 2:15:05 PM }<CRLF>

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}␣

< Back Next > Finish Cancel

4. Enter "Companies" as the DFDL Schema file name and ensure the Library name is set to the correct value (our example is using MessageModelling_TDS).

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:

< Back Next > Finish Cancel

Note that the Message name will auto complete based on the DFDL schema file name.

5. Leave the "End of record character" default value (carriage return, line feed).

Leave the "The first record is a header" and "The last record is a trailer" checked.

In the Header fields tab, enter "Header{" as the Header initiator, and set "Number of fields" = 2.

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
Header initiator: Header{
Number of fields: 2

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

< Back Next > Finish Cancel

- Click on the "Trailer fields" tab, and enter "Trailer{" as the Trailer initiator and set "Number of fields" = 1.

Click Finish.

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

Trailer initiator: Trailer{
Number of fields: 1

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

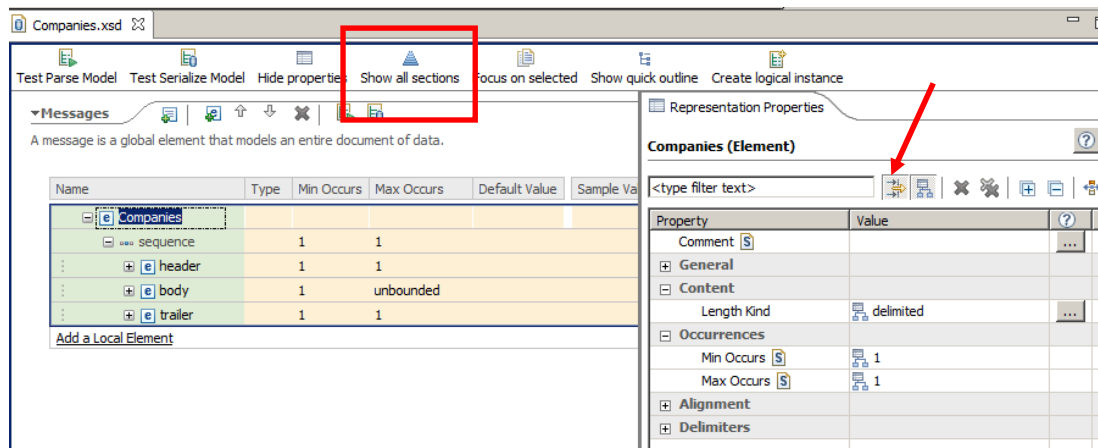
< Back Next > Finish Cancel

3. Refine the Message Model

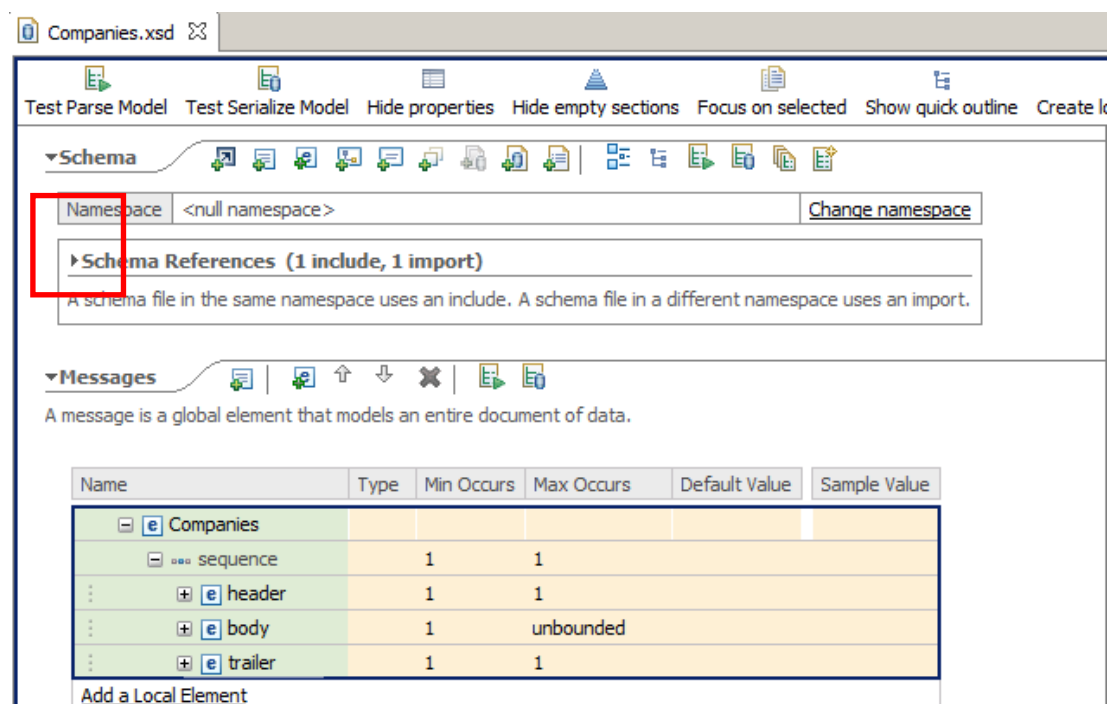
1. The DFDL editor will open with the generated DFDL Schema.

Click the "Show all sections" icon.

If you are using IIB V9.0.0.1 or earlier, you will also see the "Show Advanced" icon. You should click this icon now. In V9.0.0.2 (and in V10), this icon has been removed, and has been replaced by a small icon in the Representation Properties (see the arrowed icon below). The default is that Advanced properties are shown.



2. Click the twisty next to the "Schema References" section to expand the references (Includes and Imports) of the DFDL Schema file.



- This message model is going to build on the message model from the basic TDS lab, by creating a reference to the Company schema.

Click the "Add a reference to another schema" icon.

Companies.xsd

Test Parse Model Test Serialize Model Hide properties Hide empty sections Focus on selected Show quick outline Crea

▼ Schema

Namespace <null namespace> [Change namespace](#)

▼ Schema References + ✕

A schema file in the same namespace uses an include. A schema file in a different namespace uses an import.

Imports IBMdefined/RecordSeparatedFieldFormat.xsd http://www.ibm.com/dfdl/RecordSeparatedFieldFormat

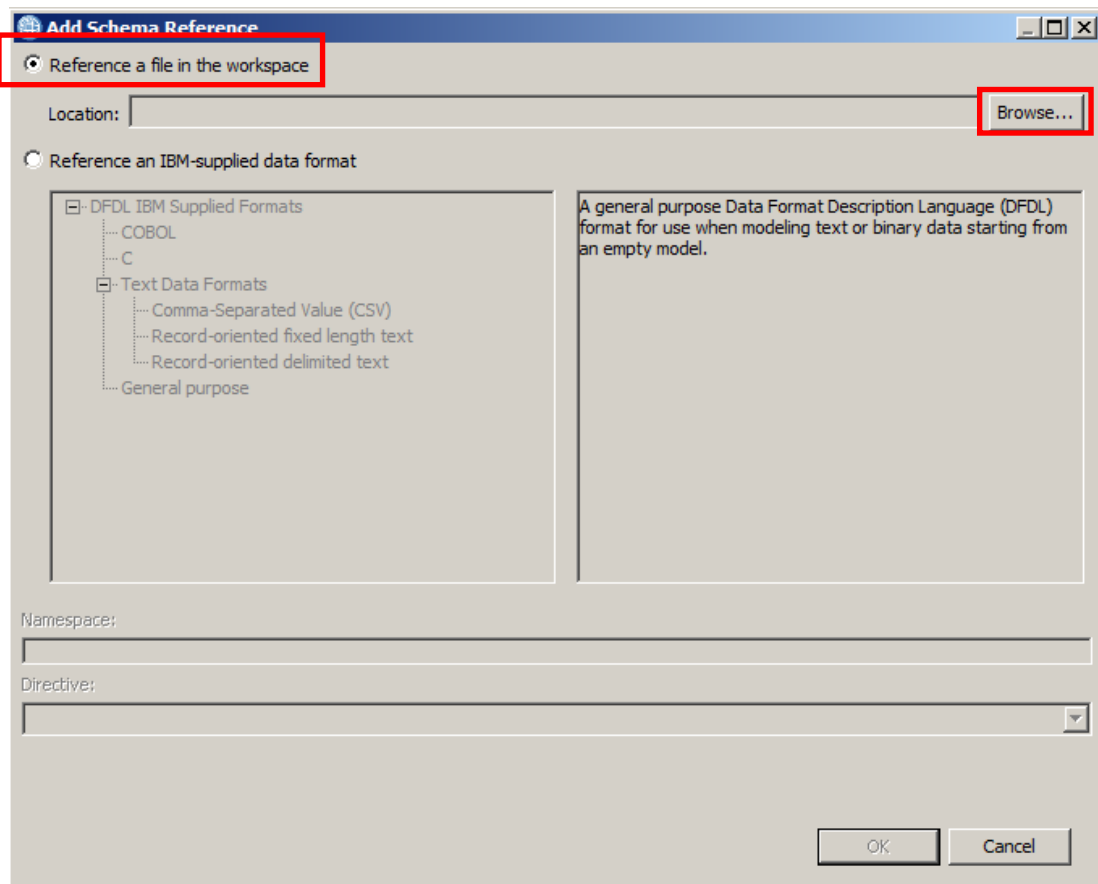
▼ Messages

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

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] [e] Companies					
[-] ... sequence		1	1		
⋮					
[+] [e] header		1	1		
⋮					
[+] [e] body		1	unbounded		
⋮					
[+] [e] trailer		1	1		

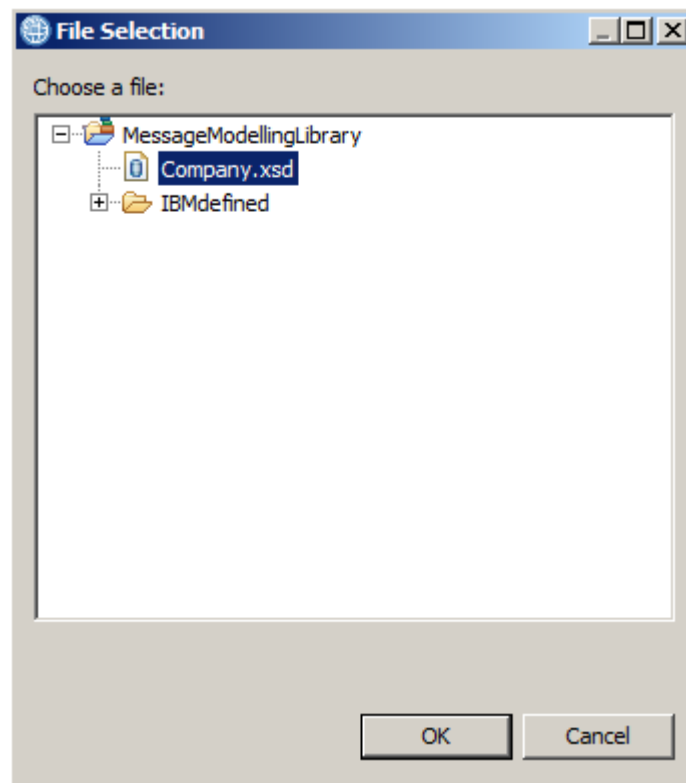
[Add a Local Element](#)

4. In the "Add Schema Reference" window, leave the default "Reference a file in the workspace" option and click Browse.



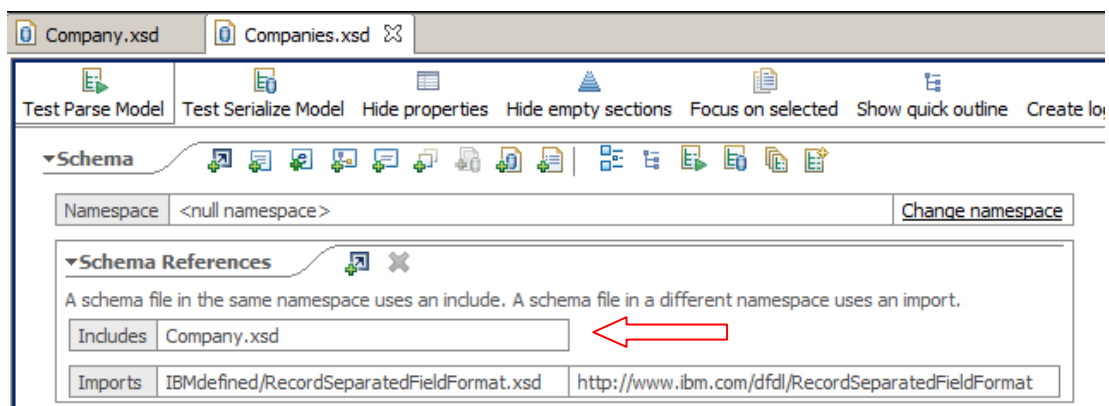
5. Select the "Company.xsd" file you created in the previous (basic) lab, or the one in the pre-built library, MessageModelling_TDS.

Click OK.



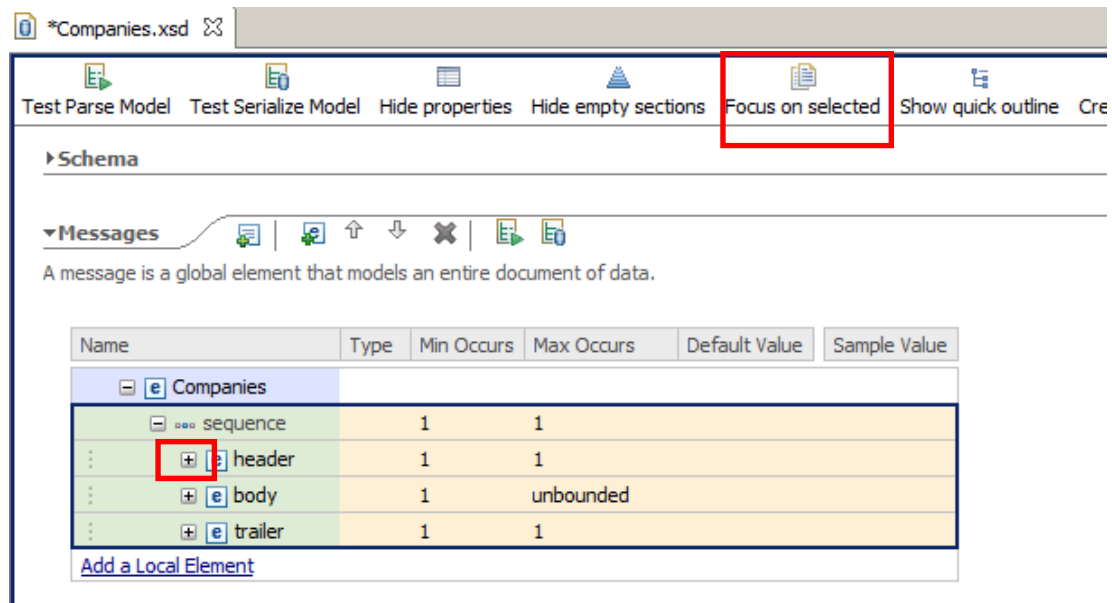
Back in the "Add Schema Reference" window click OK.

6. Note that you have two Schema References:
 - Company.xsd (which you've just added).
 - RecordSeparatedFieldFormat.xsd (which was automatically added by the wizard): This contains Record Separated specific defaults for DFDL properties. This schema already existed in the library (it is an IBM-defined schema), and Companies.xsd now has a reference to this existing schema.



- Click anywhere inside the Companies Message section, and click on the "Focus on selected" icon.

Then, expand the header element by clicking the "+".



- Change the header element's name to "Header" (capital "H") by highlighting it, and overtyping.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] [e] Companies					
[-] [e] sequence		1	1		
[+] [e] Header		1	1		
[-] [e] sequence		1	1		
[e] head_elem1	string	1	1		head_value1
[e] head_elem2	string	1	1		head_value2
[+] [e] body		1	unbounded		
[+] [e] trailer		1	1		

- Change the 2 elements under Header to "RecordDescription" and "CompanyCount".

If you wish, you can also change the sample values, as shown.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] Companies					
[-] sequence		1	1		
[-] Header		1	1		
[-] sequence		1	1		
[+] RecordDescription	string	1	1		RecDesc
[+] CompanyCount	string	1	1		CompCount
[+] body		1	unbounded		
[+] trailer		1	1		

Add a Local Element

- Delete the "body" element by right-clicking on the line of the element and selecting Delete. (Do not right-click on the text of the element name ... you will see a different context menu).

Name	Type	Min Occurs	Max Occurs
[-] Companies			
[-] sequence		1	1
[-] Header		1	1
[-] sequence		1	1
[+] RecordDescription	string	1	1
[+] CompanyCount	string	1	1
[+] body		1	unbounded

- 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**
- Add Sequence Ctrl+L, S
- Add Choice Ctrl+L, C

11. Change the trailer element's name to "Trailer" (capital "T").

Change the name of the element under Trailer to "ChkSum".

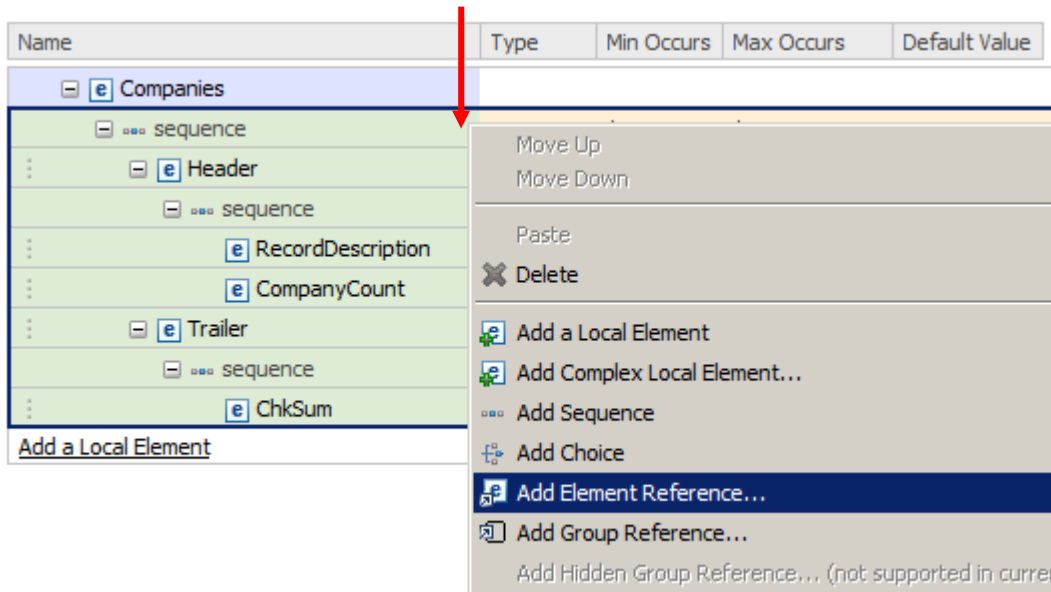
Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] e Companies					
[-] ... sequence		1	1		
⋮					
[-] e Header		1	1		
[-] ... sequence		1	1		
⋮					
[-] e RecordDescription	string	1	1		RecDesc
⋮					
[-] e CompanyCount	string	1	1		CompCount
⋮					
[-] e Trailer		1	1		
[-] ... sequence		1	1		
⋮					
[-] e ChkSum	string	1	1		checkSum

Add a Local Element

12. Earlier, you added a reference to the Company.xsd schema.

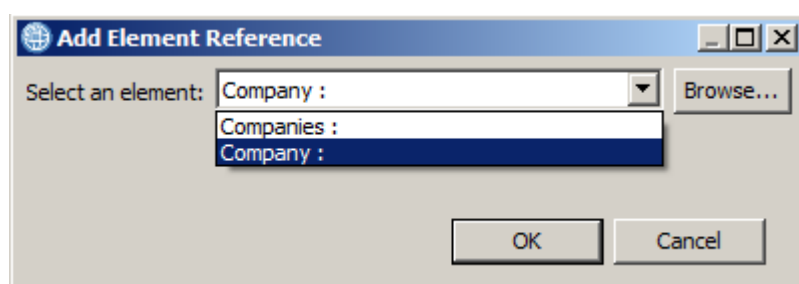
You are now going to use this reference, by adding an Element Reference in the appropriate part of the Companies message.

To do this, right-click on the Companies' "sequence" element and select "Add Element Reference". Note - you must position the mouse to the right of the word "sequence". If you are too close, a different context menu will appear.



13. Select "Company :" from the dropdown


Click OK.




14. The new element has to be positioned correctly, so right-click on the newly added element reference, and select "Move Up".

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] [e] Companies					
[-] ... sequence		1	1		
⋮ [-] [e] Header		1	1		
[-] ... sequence		1	1		
⋮ [e] RecordDescription	string	1	1		RecDesc
⋮ [e] CompanyCount	string	1	1		CompCount
⋮ [-] [e] Trailer		1	1		
[-] ... sequence		1	1		
⋮ [e] Ch	string	1	1		checkSum
⋮ [+ [f] Company					
Add a Local Element					

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

Note that the added element reference has a different icon  to differentiate from a regular element (this is only a reference to an existing element in another DFDL schema).

Also note that the element reference's name is greyed out because it is read-only. To modify it you will need to open the DFDL schema where it was defined clicking on the yellow arrow that appears when you hover over the element. ().

When you have moved the Company element, the model will look like this.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] [e] Companies					
[-] ... sequence		1	1		
⋮ [-] [e] Header		1	1		
[-] ... sequence		1	1		
⋮ [e] RecordDescription	string	1	1		RecDesc
⋮ [e] CompanyCount	string	1	1		CompCount
⋮ [+ [f] Company		1	1		
i [-] ... sequence		1	1		
[e] CompanyName	string	1	1		a
+ [e] Employee		1	unbounded		
⋮ [-] [e] Trailer		1	1		
[-] ... sequence		1	1		
⋮ [e] ChkSum	string	1	1		checkSum
Add a Local Element					

15. Expand both the Header and Trailer elements by clicking the "+" next to them.

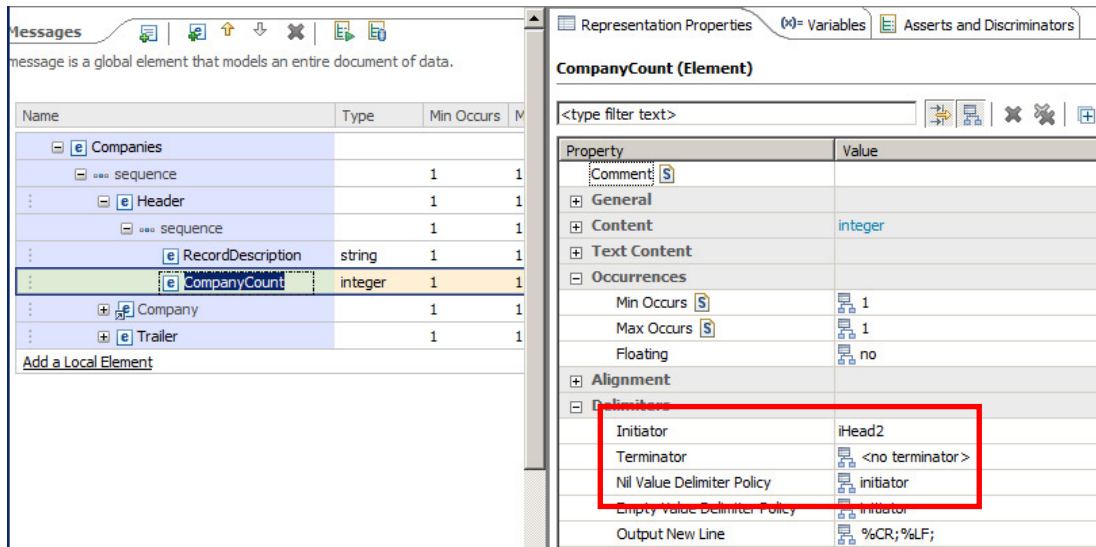
Name	Type	Min Occurs	Max Occurs	Default Value
[-] [e] Companies				
[-] ... sequence		1	1	
⋮ [e] Header		1	1	
⋮ [+][e] Company		1	1	
⋮ [e] Trailer		1	1	
Add a Local Element				

16. Change the CompanyCount element type to "integer" by clicking its type column and selecting "integer" (not int).

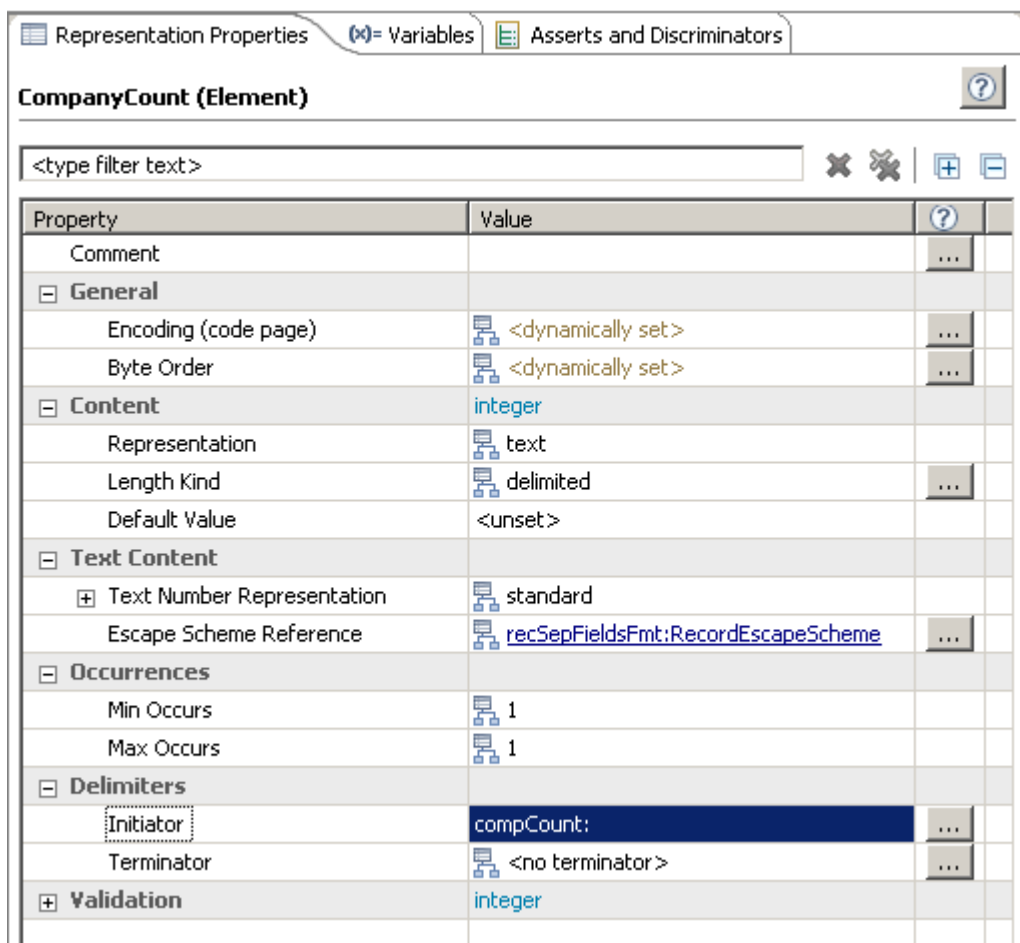
Name	Type	Min Occurs	Max Occurs	Default
[-] [e] Companies				
[-] ... sequence		1	1	
⋮ [-] [e] Header		1	1	
[-] ... sequence		1	1	
⋮ [e] RecordDescription	string	1	1	
⋮ [e] CompanyCount	integer	1	1	
⋮ [+][e] Company				
⋮ [+][e] Trailer				
Add a Local Element				

- decimal
- double
- float
- hexBinary
- int
- integer
- long
- nonNegativeInteger
- short
- string
- time

- Click the CompanyCount element again, and look at the "Delimiters" section in the Representation Properties.



- Change the Initiator's value ("iHead2") to "compCount:" (don't miss out the : (colon)).



19. Repeat the previous step for the "RecordDescription" and ChkSum fields with the following values:

RecordDescription	recDesc:
ChkSum	checksum:

Again, do not miss the colon characters. The case of the property values is important.

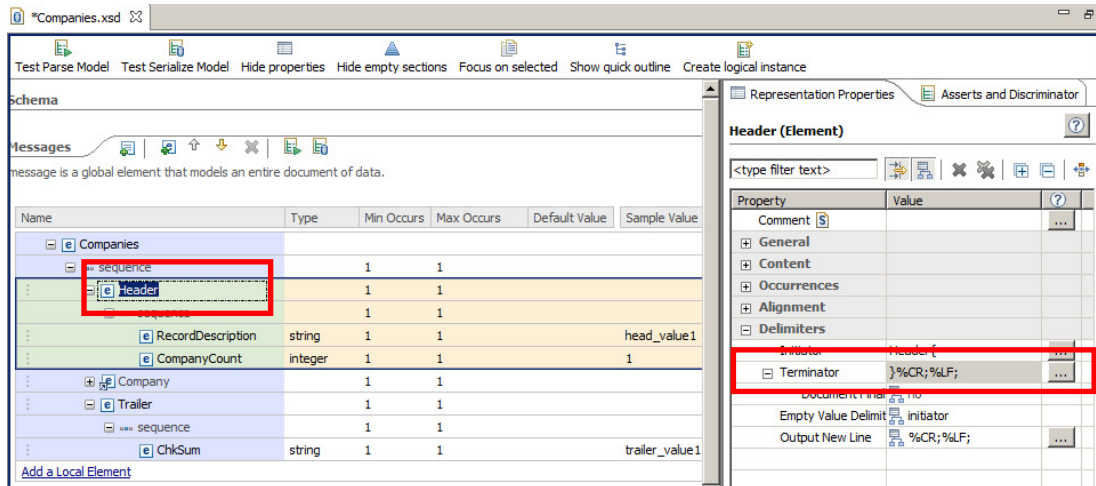
Record Description:

ChkSum (in Trailer):

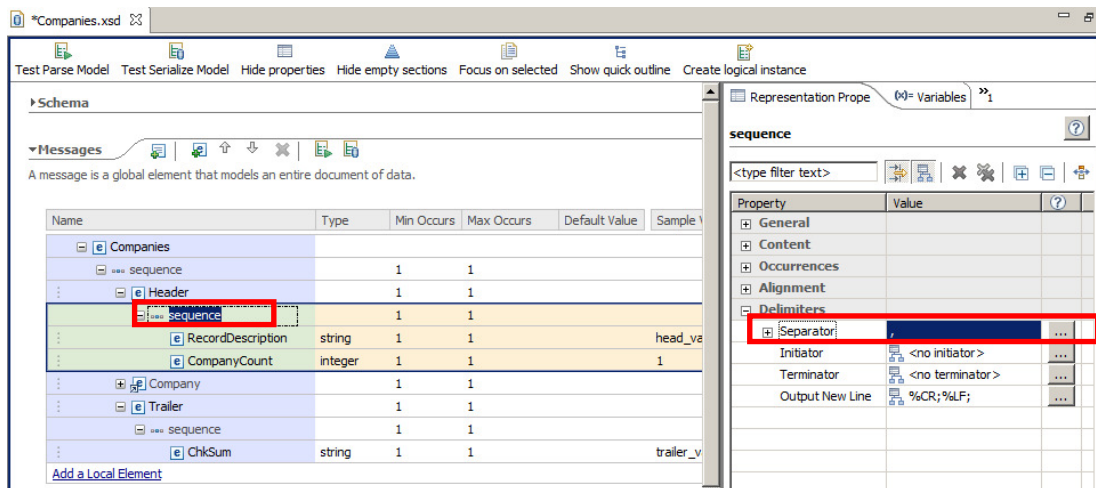
20. Now click the <sequence> content of the Companies element.

In the Representation Properties, expand the Delimiters section. Delete the value of the Separator property. Click Return to make sure the value is updated. (This is a separator automatically added by the wizard, which is not needed in this case)

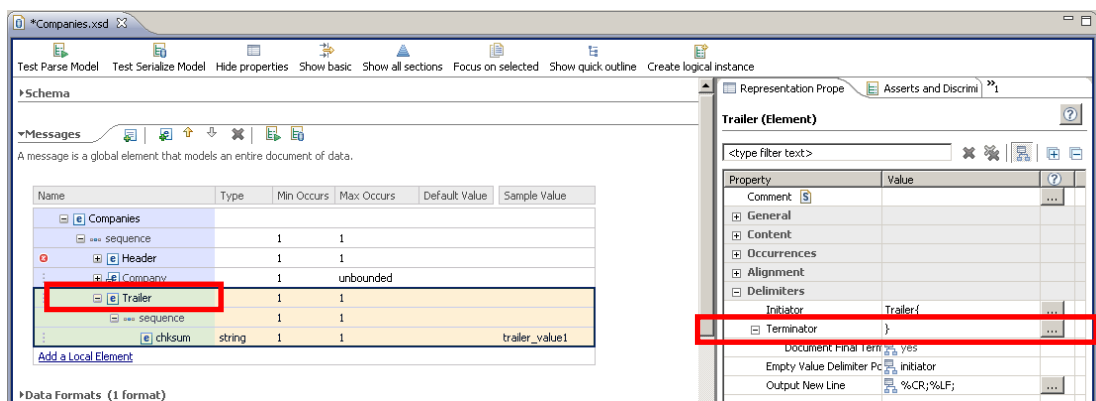
21. Click the Header element. In the Representation Properties Delimiters section, set the Terminator property to "}%CR;%LF;" (the end of the Header record is a }, followed by CRLF).



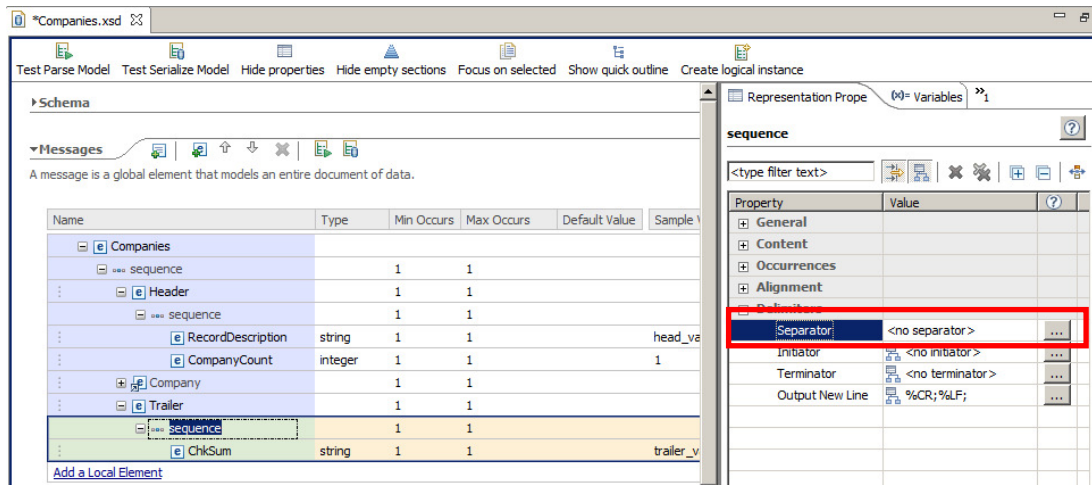
22. Then click the <sequence> content of the Header element and in the Delimiters section again, set the Separator property to "," (header elements are separated by a comma).



23. Click the Trailer element, and in the Delimiters section, set the Terminator property to "}".



- Click the <sequence> content of the Trailer element and in the Delimiters section, delete the Separator property's value.



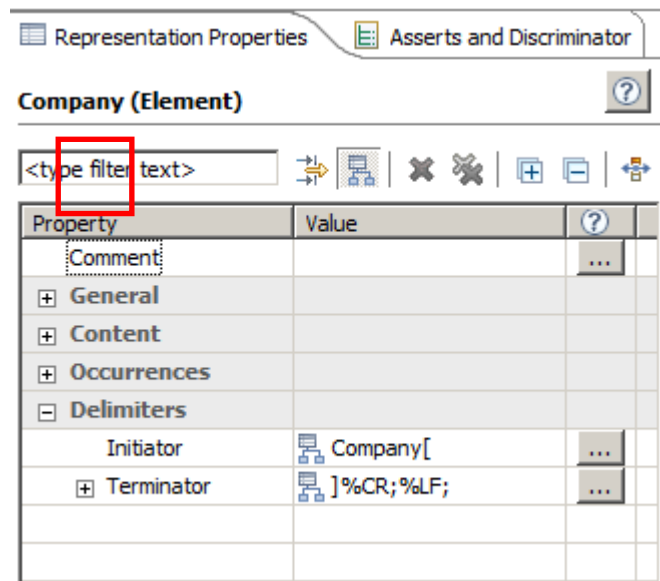
- Click the "Max Occurs" column of the Company element reference, and change it from "1" to "unbounded".

This will allow the Company element reference to have infinite occurrences.

Name	Type	Min Occurs	Max Occurs	Default Value
[-] e Companies				
[-] ... sequence		1	1	
[-] e Header		1	1	
[-] ... sequence		1	1	
[+] e RecordDescription	string	1	1	head_v...
[+] e CompanyCount	integer	1	1	1
[+] e Company		1	unbounded	
[-] e Trailer		1	1	
[-] ... sequence		1	1	
[+] e ChkSum	string	1	1	trailer_v...

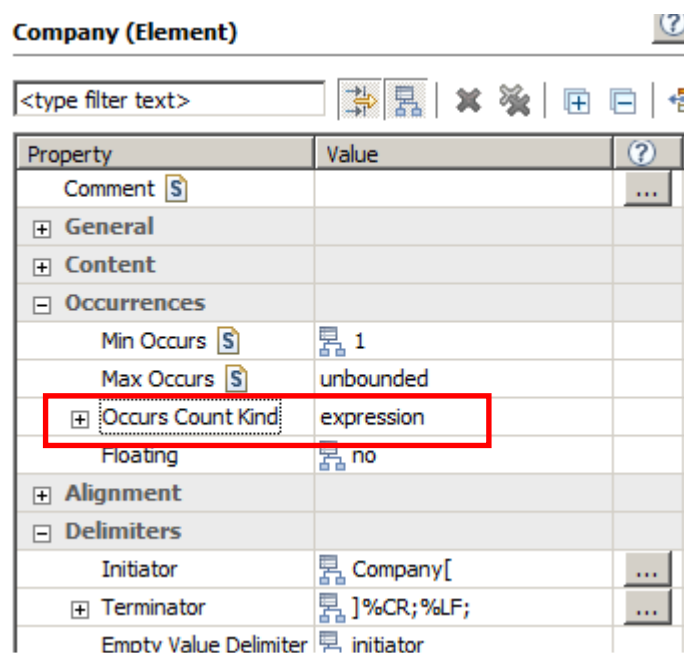
Add a Local Element

26. Click the Show all Properties button in the Representation Properties.

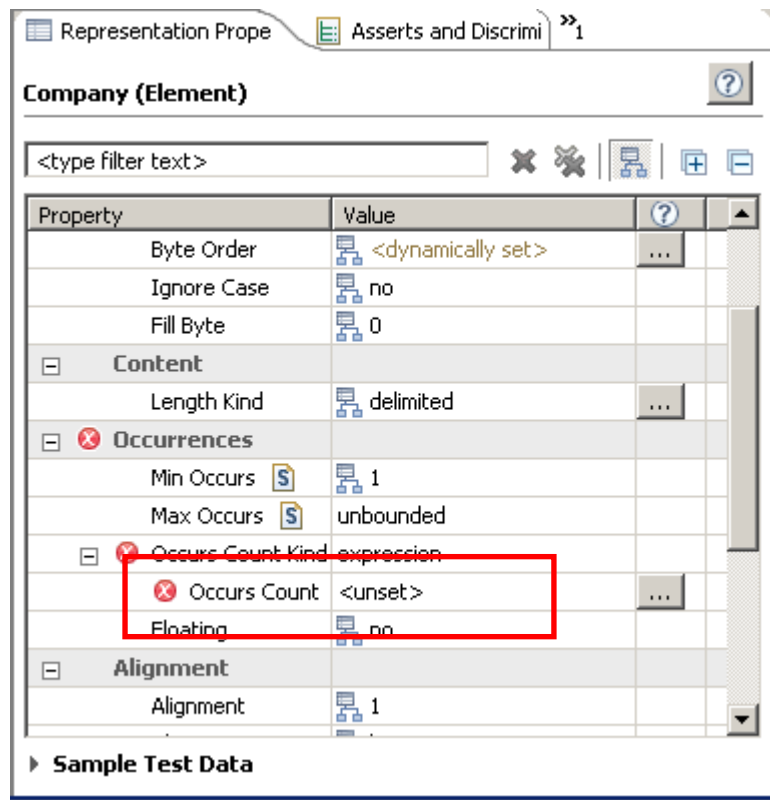


27. Click the Company element reference. In the Representation Properties Occurrences section, change the "Occurs Count Kind" from "implicit" to "expression".

You may need to Save the model and close and reopen the schema to update the Representation Properties, so that the "Occurs Count Kind" property appears.

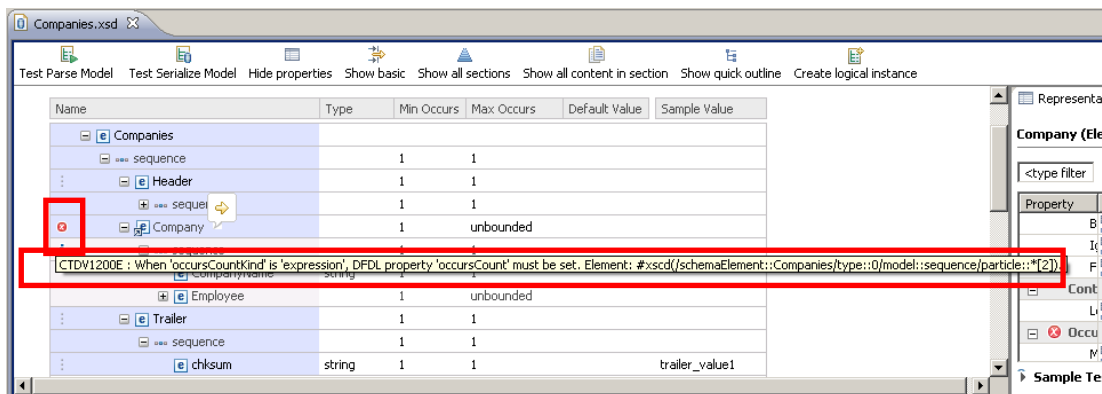


28. Expand the "Occurs Count Kind" property to show the "Occurs Count" property.
 Leave the field empty and save your XSD by pressing Ctrl+S or File->S.



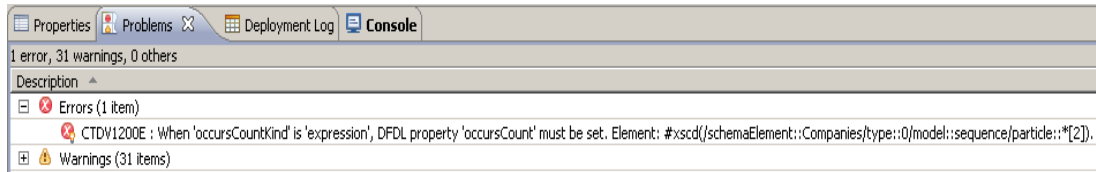
When saving, the DFCL Editor also validates the schema file.

29. Note that an error icon has appeared next to the Company element and 'Occurrences'.

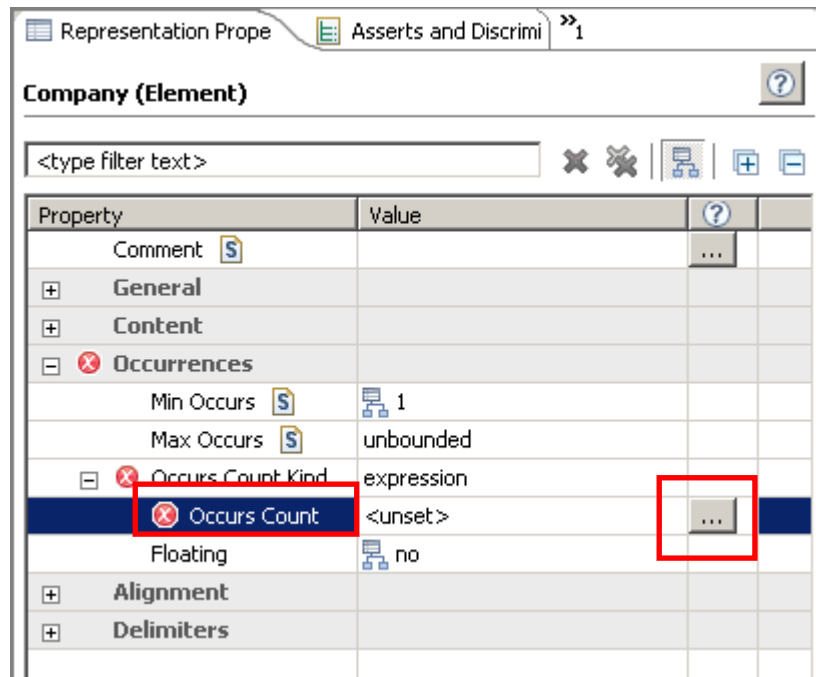


Hover over the icon to display the error description. It states that the Occurs Count property can't be empty.

30. Click the Problems tab.

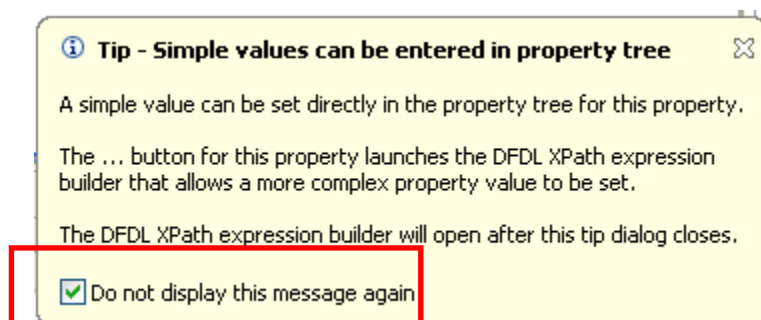


Double-click the error and the representation property of the problematic element will open.



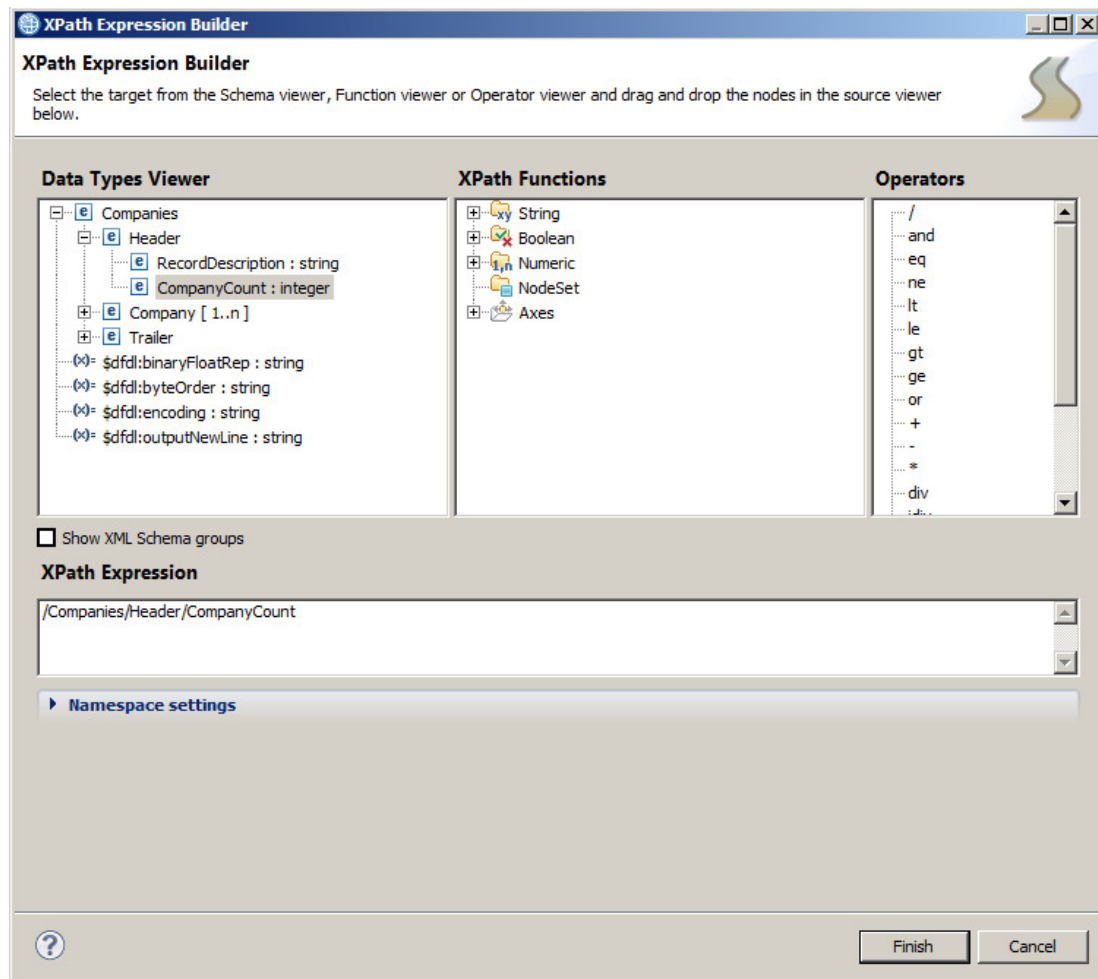
Click the button (three dots) next to the Occurs Count property.

31. A message bubble will appear. Check "Do not display this message again" and close it by clicking on the "X".



32. In the XPath Expression Builder window, expand Companies and the Header element and double-click the CompanyCount element.

Click Finish.



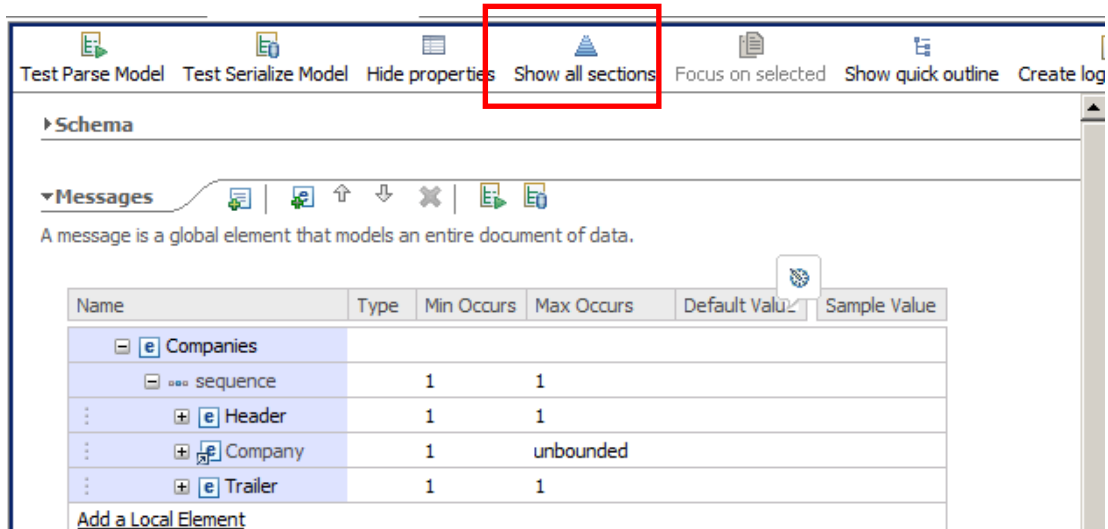
This will indicate that the number of occurrences of the Company element is dictated by the CompanyCount element in the Header.

Save the schema to clear the error.

33. Finally, the input file may, or may not, have a Carriage Return Line Feed as the last character. In some cases, the last record may be missing the final “new line” character.

To handle this situation, we will change the default property values for this model so that it will be able to handle both scenarios (ie. it will parse successfully, irrespective of whether the final character is present or not).

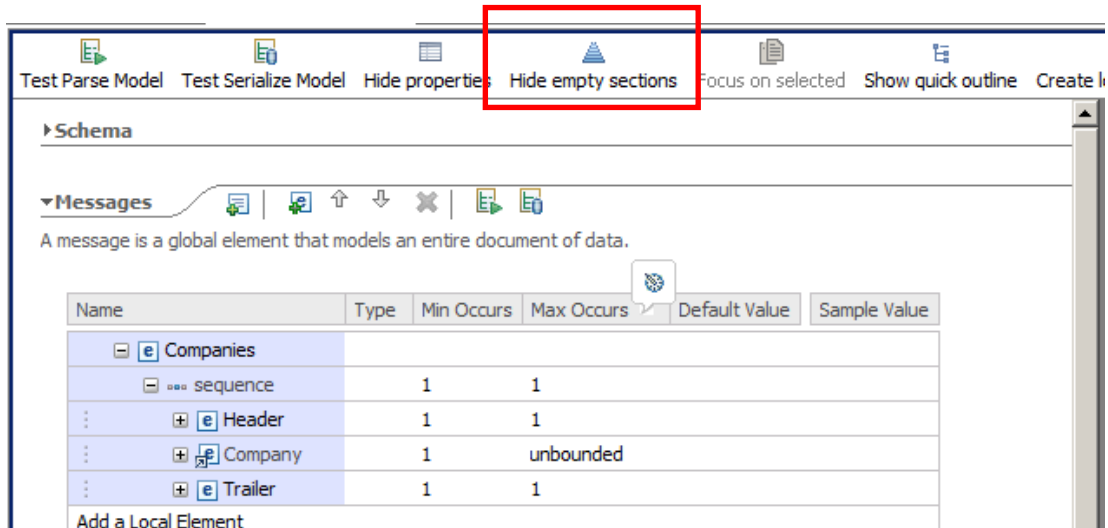
In the editor, collapse the Companies model, and then click “Show all sections”.



The screenshot shows the IBM Integration Bus V10 editor interface. The top toolbar contains several buttons: 'Test Parse Model', 'Test Serialize Model', 'Hide properties', 'Show all sections', 'Focus on selected', 'Show quick outline', and 'Create log'. The 'Show all sections' button is highlighted with a red rectangle. Below the toolbar, the 'Messages' section is expanded, showing a table of message elements. The table has columns for Name, Type, Min Occurs, Max Occurs, Default Value, and Sample Value. The 'Companies' model is expanded, showing its internal structure: a sequence, a header, a company, and a trailer.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] Companies					
[-] sequence		1	1		
[+] Header		1	1		
[+] Company		1	unbounded		
[+] Trailer		1	1		

34. To make the display a little less busy, click “Hide empty sections”.



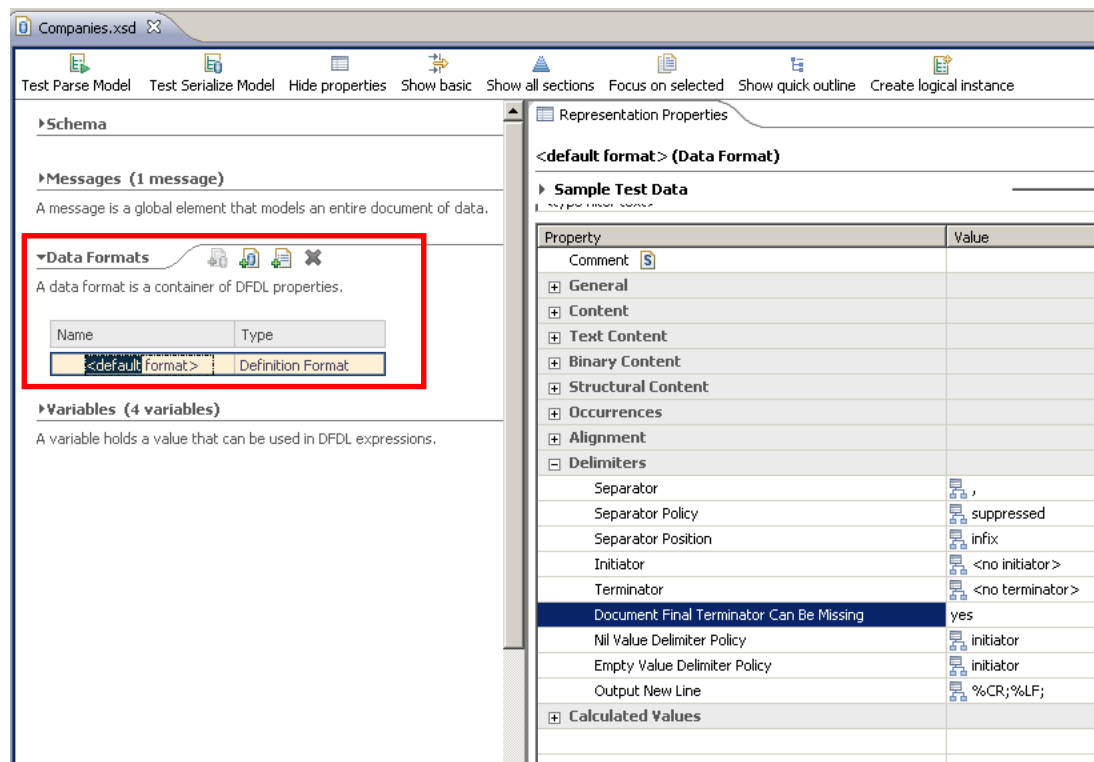
The screenshot shows the IBM Integration Bus V10 editor interface. The top toolbar contains several buttons: 'Test Parse Model', 'Test Serialize Model', 'Hide properties', 'Hide empty sections', 'Focus on selected', 'Show quick outline', and 'Create log'. The 'Hide empty sections' button is highlighted with a red rectangle. Below the toolbar, the 'Messages' section is expanded, showing a table of message elements. The table has columns for Name, Type, Min Occurs, Max Occurs, Default Value, and Sample Value. The 'Companies' model is expanded, showing its internal structure: a sequence, a header, a company, and a trailer.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] Companies					
[-] sequence		1	1		
[+] Header		1	1		
[+] Company		1	unbounded		
[+] Trailer		1	1		

35. Expand the Data Formats section.

Highlight the <default format> field. This is where you can define many default property values for the message model.

In the Representation Properties, expand the Delimiters section, and locate the property DocumentFinalTerminatorCanBeMissing. Set this property to "yes". Ensure you press the Return key to ensure the property is correctly updated.



The screenshot displays the IBM Integration Bus V10.0.0.0 interface for editing the 'Companies.xsd' schema. The left pane shows the 'Data Formats' section expanded, with a table listing the '<default format>' as a 'Definition Format'. This entry is highlighted with a red rectangle. The right pane shows the 'Representation Properties' for the selected format, with the 'Delimiters' section expanded. The 'Document Final Terminator Can Be Missing' property is highlighted in blue and set to the value 'yes'.

Property	Value
Comment	S
General	
Content	
Text Content	
Binary Content	
Structural Content	
Occurrences	
Alignment	
Delimiters	
Separator	,
Separator Policy	suppressed
Separator Position	infix
Initiator	<no initiator >
Terminator	<no terminator >
Document Final Terminator Can Be Missing	yes
Nil Value Delimiter Policy	initiator
Empty Value Delimiter Policy	initiator
Output New Line	%CR;%LF;
Calculated Values	

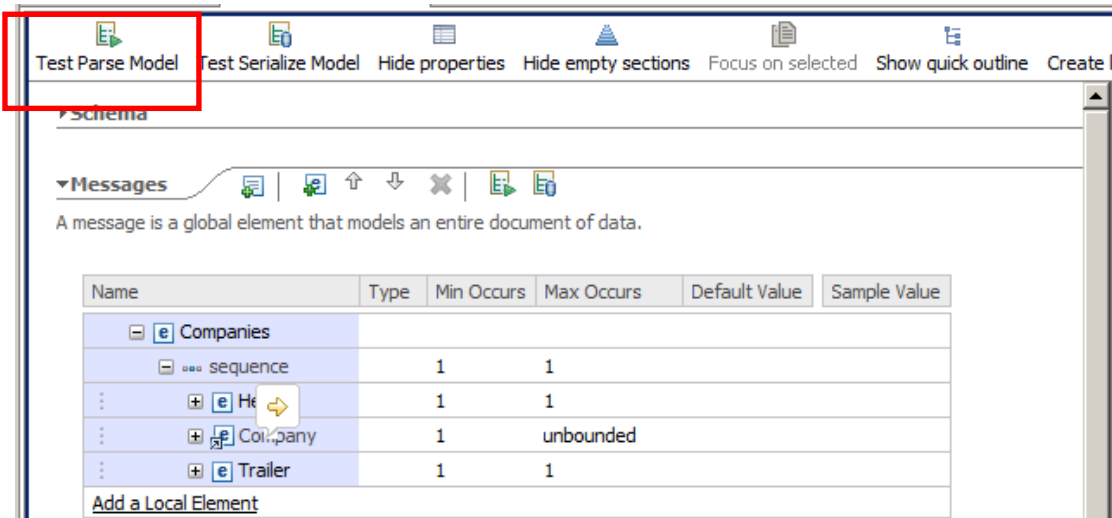
36. Press Ctrl+S or File->Save to save the schema.

Check that the error has gone and the Problems view is clean.

4. Testing the Message Model

1. Now that the Message Model is complete, you will test it against a delimited file.

Click the Test Parse Model icon.

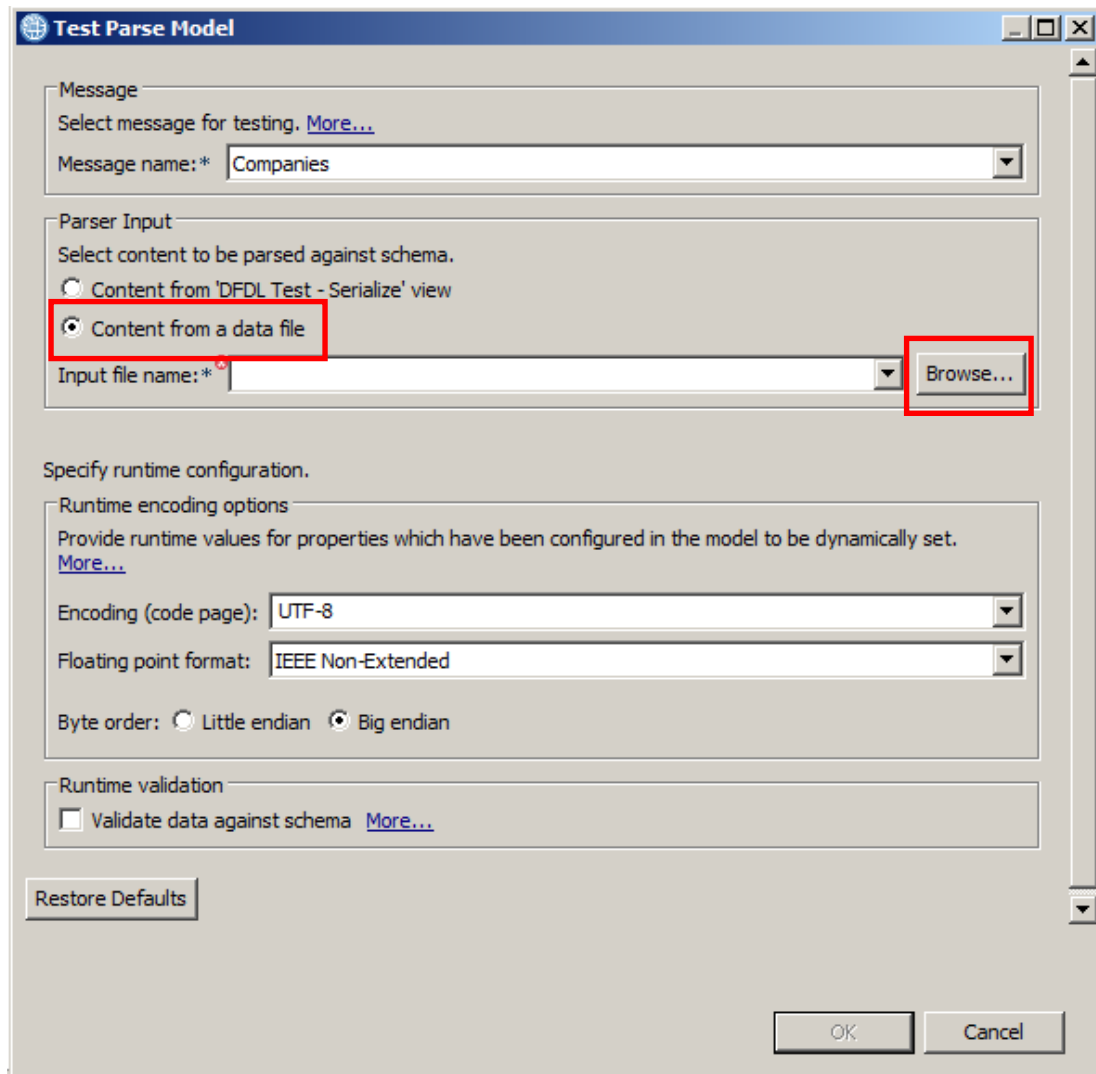


The screenshot shows the IBM Integration Bus Message Model editor interface. The top toolbar contains several icons, with the 'Test Parse Model' icon (a document with a green arrow) highlighted by a red rectangular box. Other icons include 'Test Serialize Model', 'Hide properties', 'Hide empty sections', 'Focus on selected', 'Show quick outline', and 'Create'. Below the toolbar, the 'Messages' section is expanded, showing a table of message elements. The table has columns for Name, Type, Min Occurs, Max Occurs, Default Value, and Sample Value. The elements listed are 'Companies' (with a sub-element 'sequence'), 'Header', 'Company', and 'Trailer'. The 'Header' element is currently selected, and a tooltip is visible over it. Below the table is a link that says 'Add a Local Element'.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
[-] Companies					
[-] sequence		1	1		
[+] Header		1	1		
[+] Company		1	unbounded		
[+] Trailer		1	1		

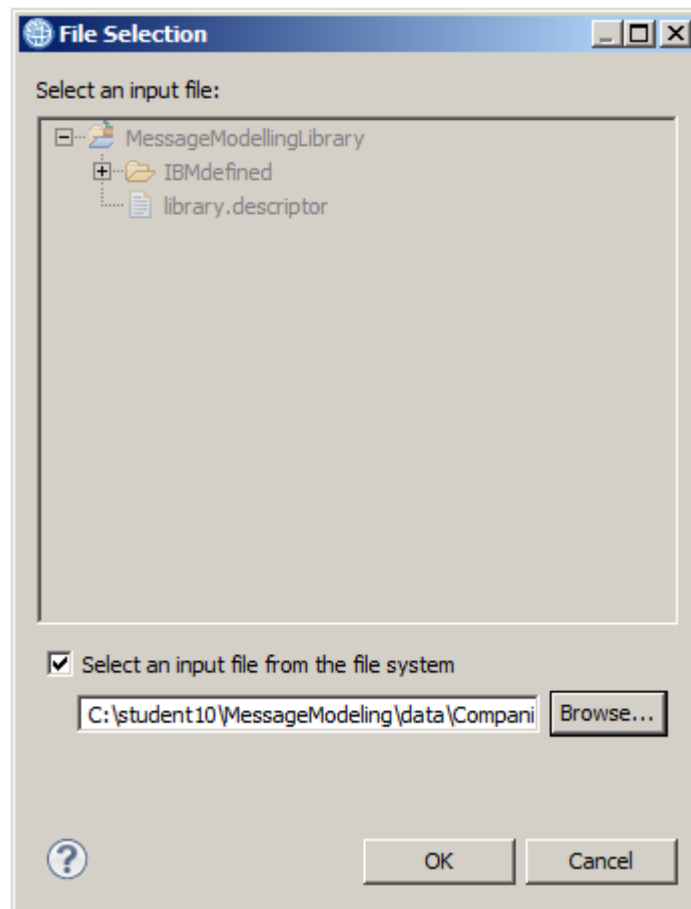
[Add a Local Element](#)

2. In the Parser Input section, select "Content from a data file" and click the Browse button.

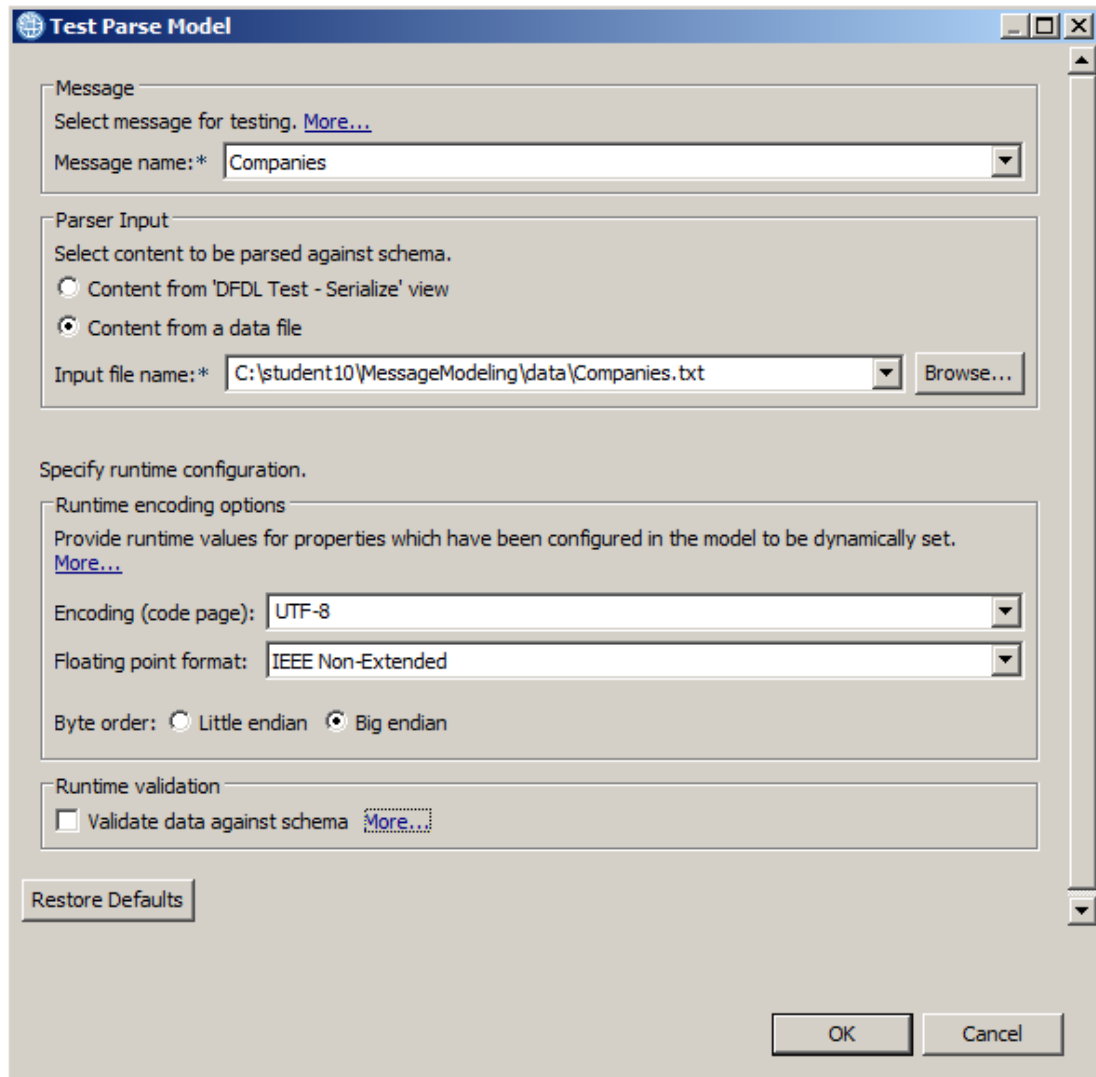


3. Check the "Select an input file from the file system" and browse to "C:\student10\MessageModeling\data" and select the "Companies.txt" file.

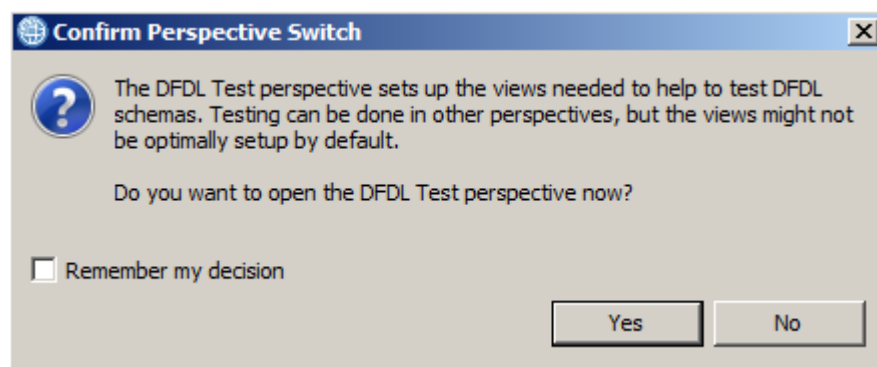
Click OK.



4. Leave all the default values and click OK.

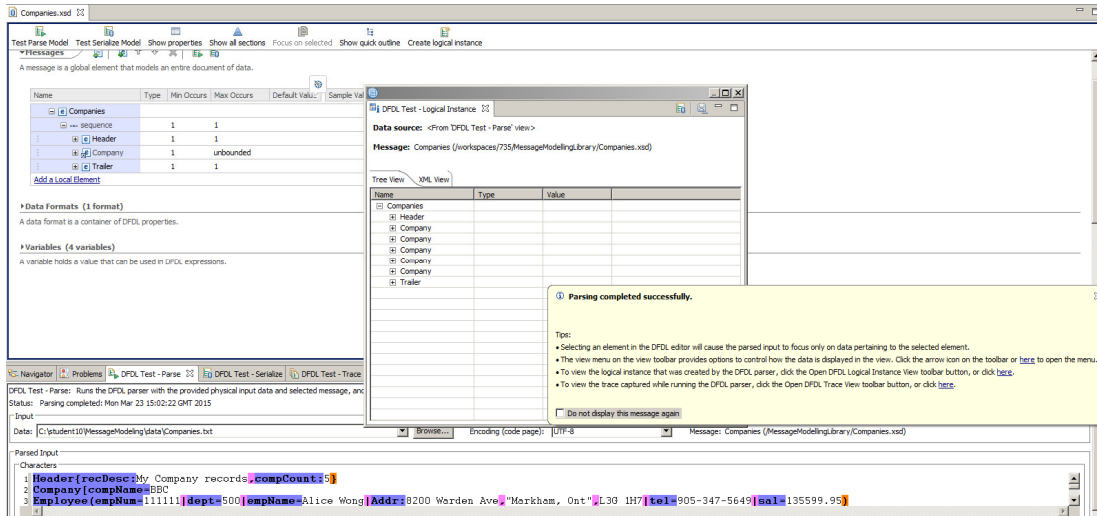


5. At the prompt, confirm the perspective switch, by clicking Yes.



- A message bubble with a "Parsing completed successfully" should appear.

Check the "Do not display this message again" and close it by clicking on the "X", or click anywhere on the test parse window.



- Review the DFDL Test - Parse and the Logical Instance views to verify the parsing was correct.

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

Message: Companies (/workspaces/735/MessageModellingLibrary/Companies.xsd)

Tree View XML View

Name	Type	Value
[-] Companies		
[-] Header		
RecordDescription	xs:string	My Company rec...
CompanyCount	xs:integer	5
[-] Company		
CompanyName	xs:string	BBC
[-] 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		
[-] Company		
[-] Company		
[-] Company		
[-] Company		
[-] Trailer		
ChkSum	xs:string	1234567890

5. Establishing repeating elements by parsing

IIB V9.0.0.2 introduced the support for the DFDL facilities to enable automatic detection of repeating elements. This is provided with the "Occurs Count Kind" property. This support is enabled when this property is set to "parsed".

As a reference, the following table summarises the different values that the "Occurs Count Kind" property can take, and the meaning of each value. Note that the corresponding DFDL property name is occursCountKind.

occursCountKind value	Meaning	minOccurs and maxOccurs used
fixed	count given by maxOccurs	To provide count (minOccurs must equal maxOccurs)
expression	count given by element in data	For validation only
implicit	count bounded by minOccurs and maxOccurs	To bound count
parsed	count unbounded	For validation only

1. The Companies.xsd model has been defined to use the "Occurs Count Kind=expression" for the Company element. The value of the property is obtained from the CompanyCount element in the Header record.

The screenshot displays the IBM Integration Bus (IIB) interface for editing the XSD model 'Companies.xsd'. The left pane shows the 'Messages' section with a tree view of the schema elements. The 'Company' element is highlighted with a red box, showing its properties: Min Occurs: 1, Max Occurs: unbounded. The right pane shows the 'Company (Element)' properties, with 'Occurs Count Kind' set to 'expression' and 'Max Occurs' set to 'unbounded', both highlighted with red boxes. The 'Occurs Count Kind' property is also highlighted with a red box in the 'Occurrences' section of the right pane.

2. Change the Occurs Count Kind property to "parsed".

This value means that the DFDL parser will keep trying to parse Company records until it gets a failure, at which point it will assume it has found all the Company records and move on to the Trailer record.

In this case, it is no longer necessary to specify the number of records with the CompanyCount element of the header record.

Save the updated model.

The screenshot shows the 'Company (Element)' configuration window in IBM Integration Bus V10. The window has two tabs: 'Representation Properties' (selected) and 'Asserts and Discriminators'. Below the tabs is a search bar containing '<type filter text>' and a toolbar with icons for navigation and editing. The main area is a table with two columns: 'Property' and 'Value'. The 'Occurs Count Kind' property is highlighted in blue, and its value is 'parsed'. Other properties include 'Min Occurs' (1), 'Max Occurs' (unbounded), 'Floating' (no), 'Initiator' (Company[), 'Terminator' (]%CR;%LF;), 'Empty Value Delimiter Policy' (initiator), and 'Output New Line' (]%CR;%LF;).

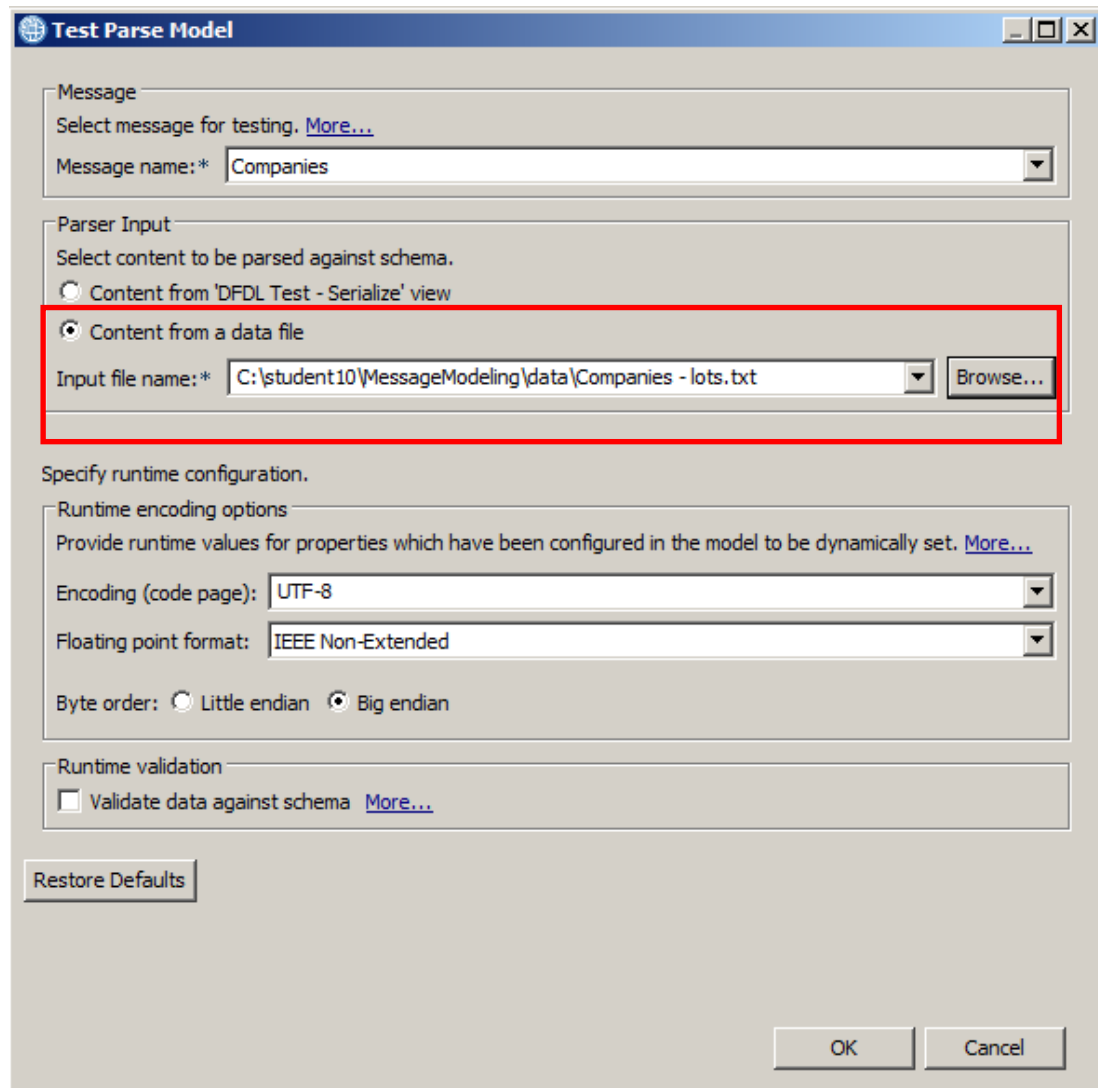
Property	Value
Comment	
General	
Content	
Occurrences	
Min Occurs	1
Max Occurs	unbounded
Occurs Count Kind	parsed
Floating	no
Alignment	
Delimiters	
Initiator	Company[
Terminator]%CR;%LF;
Empty Value Delimiter Policy	initiator
Output New Line]%CR;%LF;

3. Test the updated model.

Click Test Parse Model.

In the dialogue, select the file **c:\student10\MessageModeling\data\Companies - lots.txt**.

This file has more company records than the earlier input file. However, the company record count has been left unchanged, at 5, which does not match the number of company records in this new file.

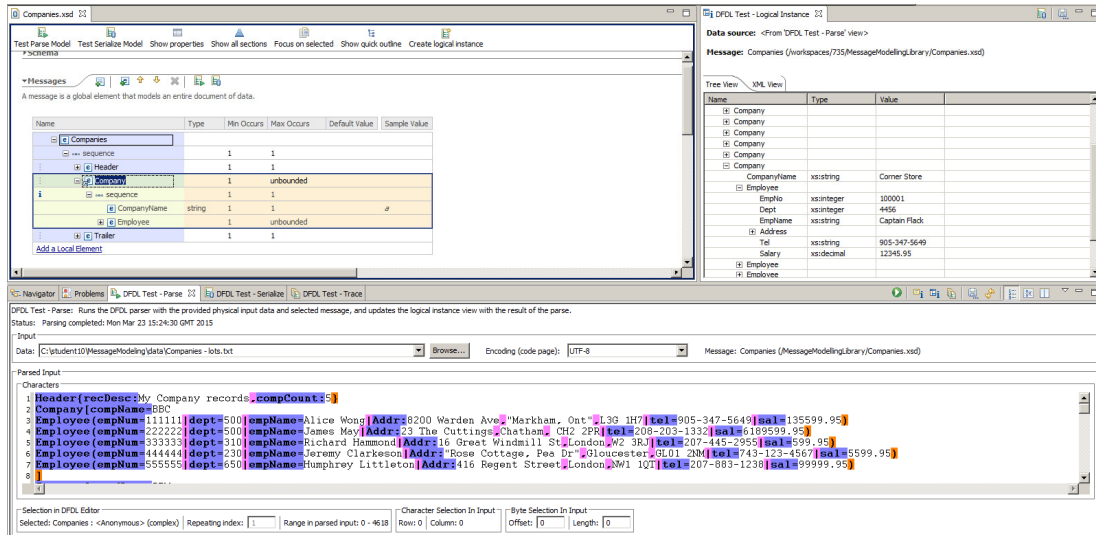


4. Click OK to invoke the Test Parser.

The new file will be parsed, and all records will be successfully displayed, even though the CompanyCount element still has the value 5.

The screen capture below shows the parsed infocset (top right pane) with all Company records.

The Test Parse pane (bottom) shows the full details of all parser records. Note the value of CompanyCount (compCount) is still 5.



5. Take a quick look at the DFDL Test - Trace tab.

Near the bottom of the trace, you will see how the parser has determined that it has found the final Company element.

First, during processing of the 10th Company element, it looks for an Employee element. Since there are no more Employee elements, the parser determines this is not available, and tries to find a new Company element.

```

info: Offset: 4589. Starting to process element 'Employee'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company/type::0/model::sequence/schemaElement::Employee), 61]

info: Offset: 4589. Did not find initiator for 'Employee'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company/type::0/model::sequence/schemaElement::Employee), 59]

info: Offset: 4589. Element 'Employee' is empty because the initiator was not found in the data stream.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company/type::0/model::sequence/schemaElement::Employee), 105]

error: CTDP3041E: Initiator 'Employee(' not found at offset '4,589' for element '/Companies[1]/Company[10]/Employee[6]'.
    
```

A little further down, it then tests for a new Company element. However, since it has processed the final Company element, the parser detects that no more are available (it could not find the initiator).

```

info: Offset: 4592. Optional element 'Company' encountered. The DFDL parser will return to this position if the e
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company), 156]

info: Offset: 4592. Starting to process element 'Company'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company), 60]

info: Offset: 4592. Did not find initiator for 'Company'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company), 58]

info: Offset: 4592. Element 'Company' is empty because the initiator was not found in the data stream.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company), 104]

error: CTDP3041E: Initiator 'Company[' not found at offset '4,592' for element '/Companies[1]/Company[11]'.
    
```


So, finally, the parser decides that the 10th occurrence of Company is the last one available, and so processes the next element as a Trailer element. This is parsed successfully, using the Trailer initiator.

```
info: Offset: 4592. Parser was unable to resolve data on the current branch and will evaluate the next available b
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Companies/type::0/model::sequence), 216]

info: Offset: 4592. Occurrence '11' of element 'Company' was not found in the data. occursCountKind is 'parsed' sc
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Company), 171]

info: Offset: 4592. Starting to process element 'Trailer'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Companies/type::0/model::sequence/schemat

info: Offset: 4592. Found initiator 'Trailer{' for 'Trailer'
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Companies/type::0/model::sequence/schemat

info: Offset: 4600. Starting to process element 'ChkSum'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Companies/type::0/model::sequence/schemat

info: Offset: 4600. Found initiator 'chksum:' for 'ChkSum'
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Companies/type::0/model::sequence/schemat

info: Offset: 4607. Found delimited value: '1234567890' for element 'ChkSum'. The delimiter was '}'.
[dfdl = /MessageModelling_TDS/Companies.xsd, scd = #xscd(/schemaElement::Companies/type::0/model::sequence/schemat
```

6. Unordered sequences

Finally, you will examine the facilities for handling unordered sequences. This was also introduced in IIB Version 9.0.0.2

This facility will be demonstrated by switching the order of two elements in the Header record of the Companies.txt file, and by setting the appropriate property in the model to use "unordered".

1. Switch back to the Application Development perspective.

In the Companies.xsd model, expand the Header element, and highlight the "sequence" underneath Header (ie. the sequence referring to the two elements contained within Header).

In the Representation Properties (Content), observe that the Sequence Kind property is set to "ordered".

The screenshot displays the IBM Integration Bus V10 Application Development perspective. The main window shows the XSD model for 'Company.xsd'. The 'Messages' pane on the left shows a tree view of the model. The 'Header' element is expanded, and the 'sequence' element underneath it is selected. The 'Representation Properties' pane on the right shows the properties for the selected 'sequence' element. The 'Content' section is expanded, and the 'Sequence Kind' property is highlighted with a red box, showing its value as 'ordered'.

Name	Type	Min Occurs	Max Occurs
Companies			
sequence		1	1
Header		1	1
sequence		1	1
RecordDescription	string	1	1
CompanyCount	integer	1	1
sequence		1	1
Company		1	unbounded
Trailer		1	1

Property	Value
General	
Data Format Reference	<default format>
Encoding (code page)	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Content	
Initiated Content	yes
Sequence Kind	ordered
Occurrences	
Min Occurs	1
Max Occurs	1
Alignment	
Delimiters	
Separator	/

- Set the property to "unordered", and save the model.

Note that you are not changing the order of the elements in the model definition.

The screenshot displays the IBM Integration Bus V10 Message Modelling tool interface. The main window shows a message model for 'Companies.xsd'. The 'Messages' pane on the left lists the elements: Companies (sequence), Header (sequence), RecordDescription (string), CompanyCount (integer), Company (sequence), and Trailer (sequence). The 'Properties' pane on the right shows the 'sequence' property for the selected element, with the 'Sequence Kind' property set to 'unordered'. A red box highlights the 'Sequence Kind' property and its value 'unordered'.

Name	Type	Min Occurs	Max Occurs
Companies	sequence	1	1
Header	sequence	1	1
RecordDescription	string	1	1
CompanyCount	integer	1	1
Company	sequence	1	unbound
Trailer	sequence	1	1

Property	Value
Data Format Reference	<default format>
Encoding (code page)	<dynamically set>
Byte Order	bigEndian
Ignore Case	no
Fill Byte	0
Initiated Content	yes
Sequence Kind	unordered
Min Occurs	1
Max Occurs	1

3. Test the updated model again.

Click Test Parse Model.

In this case, select the file

c:\student10\MessageModeling\data\Companies_lots_unordered.txt.

This file has a lot of company records. In addition, the elements in the Header record have been switched, so that the compCount element appears before the recDesc.

The image shows two screenshots. The top one is a Notepad window titled 'Companies_lots_unordered.txt - Notepad'. The text inside is as follows:

```
Header{compCount:0,recDesc:My Company records}
Company[compName=BBC
Employee(empNum=111111|dept=500|empName=Alice wong|Addr:8200 warden
Employee(empNum=222222|dept=500|empName=James May|Addr:23 The Cuttir
Employee(empNum=333333|dept=310|empName=Richard Hammond|Addr:16 Grea
Employee(empNum=444444|dept=230|empName=Jeremy Clarkson|Addr:"Rose
Employee(empNum=555555|dept=650|empName=Humphrey Littleton|Addr:416
]
```

The bottom screenshot is the 'Test Parse Model' dialog box. It has several sections:

- Message:** 'Select message for testing. [More...](#)' and a dropdown menu with 'Companies' selected.
- Parser Input:** 'Select content to be parsed against schema.' with two radio buttons: 'Content from 'DFDL Test - Serialize' view' (unselected) and 'Content from a data file' (selected).
- Input file name:** A text box containing 'C:\student10\MessageModeling\data\Companies_lots_unordered.txt' and a 'Browse...' button. This section is highlighted with a red rectangle.
- 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' (unselected) and 'Big endian' (selected)
- Runtime validation:** 'Validate data against schema [More...](#)' (checkbox is unchecked)
- 'Restore Defaults' button
- 'OK' and 'Cancel' buttons at the bottom right.

Select OK.

- You will see that the new file has been successfully parsed, even though the Header elements in the input message have been switched round.

The screenshot shows the DFPL Test - Logical Instance window in IBM Integration Bus V10. The window is divided into several panes:

- Schema Table:** A table listing the schema elements for 'Companies.xsd'. The columns are Name, Type, Min Occurs, Max Occurs, Default Value, and Sample Value.

Name	Type	Min Occurs	Max Occurs	Default Value	Sample Value
Companies	sequence	1	1		
Header	sequence	1	1		
RecordDescription	string	1	1		head_value1
CompanyCount	integer	1	1		z
Company	sequence	1	1		
CompanyName	string	1	1		a
- Tree View:** A hierarchical view of the logical instance. The root is 'Companies', which contains 'Header', 'RecordDescription', 'CompanyCount', 'Company', 'Employee', and 'Salary'.
- Parsed Input:** A text area showing the raw XML data. The header is:


```
Header{compCount:0, recDesc:My Company records:}
Company{compName=BBC
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=6189599.95}
Employee{empNum=333333|dept=310|empName=Richard Hammond|Addr:16 Greet Windmill St,London,W2 3RJ|tel=207-445-2955|sal=999.95}
Employee{empNum=444444|dept=200|empName=Jeremy Clarkson|Addr:7 Rose Cottage, Pea Dr,Gloucester,GL01 2NW|tel=743-123-4567|sal=5599.95}
Employee{empNum=555555|dept=650|empName=Timphrey Littleton|Addr:416 Regent Street,London,W1 1QT|tel=207-883-1230|sal=99999.95}
Company{compName=TFM
```

END OF LAB GUIDE