

IBM Integration Bus

Message Modeling with DFDL

Lab 5 Using DFDL length prefixes

June 2015 Hands-on lab built at product Version 10.0.00

1.	INTF	RODUCTION	3
	1.1 1.2	LAB PREPARATION	
2.	IMPO	ORT THE BASE MODEL	1
3.	CRE	ATE THE PREFIX LENGTH CHARACTER SCENARIO	3
4.	CRE	ATE THE PREFIX LENGTH BINARY SCENARIO20)
EN	D OF	LAB GUIDE	3

1. Introduction

Support for length prefixes in the DFDL Message Modelling tools was introduced in WebSphere Message Broker V8.0.0.1 and is included in IBM Integration Bus V9.0 and V10.

A common form of data formatting uses the approach of having a prefix to the main element, where the prefix contains the length of the element itself. This capability is commonly used in message modeling, and is a particular requirement for certain types of industry standard models, for example the ISO8583 standard used in credit card processing, and the PL/1 var char type.

There are many variations of this approach. The value held in the length prefix might represent just the length of the element to which it refers, or the value in the length prefix might include the length of the prefix as well as that of the element. The length prefix itself might have different characteristics from the element, for example it may be a binary prefix whereas the element is text. It is even possible for a length prefix to have its own length provided by another length prefix!

This lab will illustrate some of these variations of length prefix specifications.

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

This lab extends the Tagged / delimited lab, and includes the new message modeling capability for prefix length fields introduced in WMB V8 Fixpack1.

The starting point for this lab is a tagged-delimited message model, with a schema definition named Company.xsd. You will create two new message models based on this, as follows:

CompanyAddressChar.xsd – some of the elements will be changed to use a 2-byte length prefix of type "character".

CompanyAddressBin.xsd – some of the elements will be changed to use a 2-byte length prefix of type "binary".

2. Import the base model

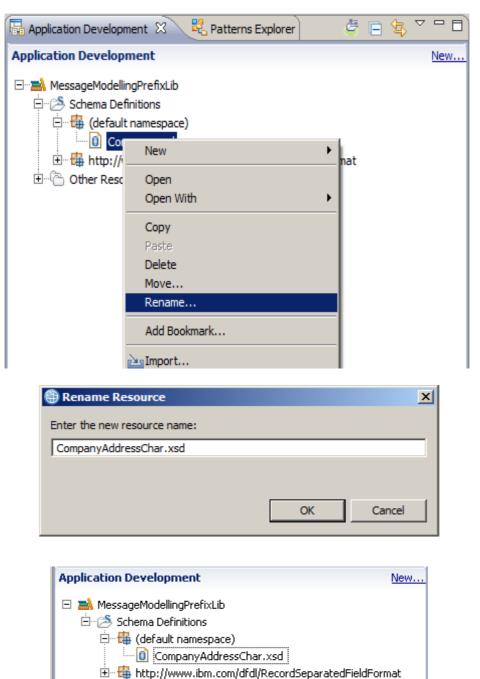
You are going to create two message models. One will use a length prefix in character form, and one will use a length prefix in binary form. Both length prefixes will be two bytes.

Both message models will be defined in the same library, so you will need to make various adjustments to the schema and message names to avoid naming conflicts.

1. Import the PI file c:\student10\MessageModelling\prefixes\PrefixLabStartingPoint.zip.

	IBM Integration Toolkit - C:\workspaces\mm5	_ 🗆 ×
File Edit Navigate Search Pro		Quick Access
	☆・○・♀・ ∦ • ∦ • ∀ • → ↓	Quick Access
Application Dev 🔀 💐 Pat	terns Explorer C	
Application Development	The second	1
New Application	Import Projects Import Projects from a zip file.	
New Integration Service New REST API		
New Library		
	From zip file: C:\student10\MessageModeling\prefixes\PrefixLabSt	
	Project location root: C:\workspaces\mm5 Browse	
	MessageModellingPrefixLib	
🖁 Int 🔀 🞇 Int 🗞 Dat.		ment Log 🛃 🚼 🛟 🗵 🔽 🗖
Canter of Conter of order		
Integration Nodes	Select All Deselect All Select Referenced	
É- 🖓 IB10NODE		
	Cancel	

2. Rename the schema Company.xsd to CompanyAddressChar.xsd.



3. Create a new copy of the schema, and call it CompanyAddressBin.xsd.

Use Ctrl-C / Ctrl-V.

Name Conflict	×
Enter a new name for "CompanyAddressChar.xsd"	
CompanyAddressBin.xsd	
OK Cancel	

4. At this point, the navigator will show several errors. This is because the two models have a global element with the same name, which is not permitted within a single library.

🔚 Application Development 🛛 🖧 Patterns Explorer	Ë	□ \$	
Application Development			New
MessageModellingPrefixLib Schema Definitions (default namespace) CompanyAddressBin.xsd CompanyAddressChar.xsd Multip://www.ibm.com/dfdl/RecordSeparated Other Resources	FieldFormat		

5. To rectify this, open the CompanyAddressBin.xsd, and in the message model editor, change the global element name to CompanyAddressBin.

0 CompanyAddressBin.xsd 🛛							
Test Parse Model Test Serialize Mode	l Hide prop	erties Show	all sections Foc.				
Messages 🛛 📮 🕆 🤄 💥 📙 📴 message is a global element that models an entire document of data.							
Nane	Туре	Min Occurs	Max Occurs				
🖃 🖻 CompanyAddressBir							
sequence		1	1				
E CompanyName	string	1	1				
Employee		1	unbounded				
Add a Local Element							

6. Saving this change (Ctrl-S) will remove the errors.

For consistency, make a similar change to the second schema, CompanyAddressChar.xsd, renaming the global element to CompanyAddressChar.

OmpanyAddressChar.xsd ⋈										
Test Parse Model Test Serialize M	iodel	Hide prope	erties Sho	A all sections	Focu					
Messages 🛛 📮 🕯 🕆	Ŷ	X E	-							
message is a global element that mo	dels	an entire do	cument of a	lata.						
Na <mark>ne</mark>		Туре	Min Occurs	Max Occurs	5					
e CompanyAddressChar]									
🖃 🚥 sequence			1	1						
: e CompanyName		string	1	1						
: 🗉 🖻 Employee			1	unbounded						
Add a Local Element										

You will now have two message models in the library, with different global elements. You are now ready to define the length prefixes.

3. Create the Prefix Length Character scenario

Open and expand the CompanyAddressChar.xsd message model.

Parse Model Test Serialize Model Hide proper	ties Show all s	ections Focus	on selec	ted Show quick outline Create logical inst	ance
essages 🛛 😹 🕆 🖟 💥 📗	E , E		^	Representation Properties	
essage is a global element that models an entir	e document of o	lata.		CompanyAddressChar (Element)	
Name	Туре	Min Occurs	Max	ŀ	📑 🛃 🗶 🔆 🗉 🖻
CompanyAddressChar				Property	Value (?
🖃 🚥 sequence		1	1	Comment S	
e CompanyName	string	1	1	General	
Employee		1	unbo	Data Format Reference	<pre><default format=""></default></pre>
🖃 🚥 sequence		1	1	Encoding (code page)	棍 <dynamically set=""></dynamically>
e EmpNo	integer	1	1	Byte Order	🛃 <dynamically set=""> 🔜</dynamically>
e Dept	integer	1	1	Ignore Case	튐 no
e Empname	string	1	1	Fill Byte	昂 0
E e Address		1	1	Content	
		1	1	Length Kind	뤔 delimited
e StreetName	string	1	1	Occurrences	
e City	string	1	1	Min Occurs S	晃 1
e ZipCode	string	1	1	Max Occurs S	晃 1
e Tel	<string></string>	1	1	Alignment	
	decimal			Delimiters	
e Salary Add a Local Element	uecimai	1	1	Initiator	Company[
Aut a Local Element				Terminator ■]%CR;%LF;
				Empty Value Delimiter Policy	昂 initiator
				Output New Line	器 %CR;%LF;

2. Highlight the Address sequence element. You will see that the separator has been set to ',' (comma); this means that all fields in the Address element are separated by commas.

This is the part of the model that we will change.

0 CompanyAddressChar.xsd 🔀						-	- 8
Test Parse Model Test Serialize Model Hide properti	es Show all s	sections Focus	in selection	ted Show quick outline Create logical insta	ance		
▼Schema 🛛 💭 💭 💭 💭 💭) 🔊 🗐	E 6 E	E 0 ^	Representation Properties 🛛 🗱 Varia	ables 🗄 Asserts and Discriminators		
						9	?
Namespace <null namespace=""></null>			-	sequence			
Schema References (0 includes, 1 impor				<type filter="" text=""></type>	🔅 🛼 🗱 🗱 🗱		-
A schema file in the same namespace uses an inc	ude. A schema	a file in a differe	ent na	Property	Value	(?)	
				🖃 General			
•Messages 🔄 😨 🕆 🦊 💥	E. E		_	Data Format Reference	<default format=""></default>		
A message is a global element that models an entire	document of a	data		Encoding (code page)	暑 <dynamically set=""></dynamically>		
				Byte Order	<pre>dynamically set></pre>		
Name	Type	Min Occurs	Max	Ignore Case	昂 no		
	Type	Min Occurs	Max	Fill Byte	昇 0		
e CompanyAddressChar				Content			
sequence		1	1	Initiated Content	昇 no		
e CompanyName	string	1	1	Sequence Kind	🛃 ordered		
😑 😑 Employee		1	unbc	Occurrences			
🖃 🚥 sequence		1	1 -	Min Occurs S	暑 1		
EmpNo	integer	1	1	Max Occurs S	晃 1		
e Dept	integer	1	1	Alignment			
e Empname	string	1	1	Delimiters	1		
E e Address		1	1	Separator	昂, <		
🖃 🚥 sequence		1	1	Initiator	🛃 <no initiator=""></no>		
: e StreetName	string	1	1	Terminator	暑 <no terminator=""></no>		
E City	string	1	1	Output New Line	暑 %CR;%LF;		
E ZipCode	string	1	1				
e Tel	<string></string>	1	1				
e Salary	decimal	1	1]			

3. You will change the elements in the Address global element to be identified and parsed by using length prefixes, instead of being comma-delimited.

In this model, the length prefix is a two-character text number.

In this case the Address global element may have a value something like this:

Addr:158200 Warden Ave14"Markham, Ont"07L3G 1H7

The StreetName field has a value of '8200 Warden Ave', and has a prefix length of 15. The City field has a value of '"Markham, Ont"', and has a prefix length of 14. The ZipCode field has a value of 'L3G 1H7', and has a prefix length of 07.

Note that the prefix length values are normal display characters, and hence can be read in clear text.

4. To define this type of model, you first need to define a Simple Type. This is used to define the physical characteristics of the prefix length. An element which has a prefix length then simply refers to the simple type.

CompanyAddressChar.xsd						-	
Test Parse Model Test Serialize Model Hide propertie	A Show all s	ections Focu	s on selec	ted Show quick outline Create logical ins	stance		
▼Schema 🗿 📮 📮 📮 💭		E 6 6	<u> </u>	Representation Properties (X)= Va	riables 🗄 Asserts and Discriminators		
					· · ·	6	?
Namespace <null namespace=""></null>			_	sequence			
Schema References (0 includes, 1 import				<type filter="" text=""></type>	🔅 🛃 🗙 💥		-
A schema file in the same namespace uses an inclu	ude. A schema	file in a differ	ent na	Property	Value	(?)	
				🖃 General			
▼Messages 🛛 📳 😰 🕆 👯 🗶	E En			Data Format Reference	<pre><default format=""></default></pre>		
A message is a global element that models an entire	document of d	ata.		Encoding (code page)	昂 <dynamically set=""></dynamically>		
				Byte Order	🛃 <dynamically set=""></dynamically>		
Name	Type	Min Occurs	Max	Ignore Case	昂 no		
	Type	Plin Occurs	Pilux	Fill Byte	暑 0		
e CompanyAddressChar				Content			
sequence		1	1	Initiated Content	튐 no		
e CompanyName	string	1	1	Sequence Kind	昂 ordered		
Employee		1	unbo	Occurrences			
sequence		1	1 -	Min Occurs S	昂 1		
e EmpNo	integer	1	1	Max Occurs S	暑 1		
e Dept	integer	1	1	Alignment			
Empname	string	1	1	Delimiters			
E e Address		1	1	Separator	昂,		
🖃 🚥 sequence		1	1	Initiator	暑 <no initiator=""></no>		
e StreetName	string	1	1	Terminator	R <no terminator=""></no>		
E City	string	1	1	Output New Line	暑 %CR;%LF;		
E ZipCode	string	1	1				
: e Tel	<string></string>	1	1				
e Salary	decimal	1	1				

Click "Show all sections" on the main editor line.

5. In the main editor pane, expand Simple Types, and then click the "Add Simple Type" button.

		Simple Types sin ple type defines t	F The allowe	X d values for one or more simple elements.
--	--	---	-------------------	--

6. In the dialogue window, set Name = TwoCharsText (you can define your own descriptive name for this type), and set "Inherit from" to "short". Click OK.

🌐 Add Simp	ple Type	
Name:	TwoCharsText	
Inherit from:	short	•
	ОКС	ancel

7. To make the editor clearer, click "Hide empty sections".

*CompanyAddressChar.xsd			_			-
Test Parse Model Test Serialize Model Hide properti	s Hide emp	ty sections	For us on s	elected Show quick outline Create logica	l instance	
: e CompanyName	string	1	1	Representation Properties 🕪= Va	ariables	
Employee		1	unbc	_		?
🖃 🚥 sequence		1	1	TwoCharsText (Type)		U
EmpNo	integer	1	1	<type filter="" text=""></type>	🔅 🔜 🗙 🖗 🛛	
e Dept	integer	1	1		- Frank Frank ()	
Empname	string	1	1	Property	Value	(?)
E e Address		1	1	Comment S		
🖃 🚥 sequence		1	1	General		
e StreetName	string	1	1	Data Format Reference	<default format=""></default>	
: e City	string	1	1	Encoding (code page)	R <dynamically set=""></dynamically>	
e ZipCode	string	1	1	Byte Order	<pre></pre>	
e Tel	<string></string>	1	1	Ignore Case	屠 no	
e Salary	decimal	1	1	Fill Byte	暑 0	
Add a Local Element				Content	short	
				Representation	🖥 text	
Global Elements (0 elements)				Length Kind	🛃 delimited	
A global element represents a named instance of a c		ala kuna		Text Content		
A global element represents a named instance of a c	omplex or sim	pie type.		Number Representation	🛃 standard	
Complex Types (0 complex types)				Number Justification	暑 right	
				Number Pad Character Pad Kind	屠 %SP;	
A complex type defines the elements and groups that	at represent a	structure.			Pa none	
				Trim Kind	昂 padChar	
▼Simple Types 📮 🗱				Escape Scheme Reference	RecSepFieldsFmt:RecordEscapeScheme	
A simple type defines the allowed values for one or r	more simple el	ements.		Alignment		
News Deep Trees				Delimiters		
Name Base Type				Initiator	R <no initiator=""></no>	
TwoCharsText short				Terminator	<	
4			- F	Empty Value Delimiter Policy	界 initiator	

8. Highlight the new Simple Type, TwoCharsText. You will see that various properties have been set for this new type, shown in the Representation Properties in the right hand pane. Some of these properties must be changed to reflect the nature of our prefix length values.

st Parse Mode			sections I	Focus on select	ted Show quick outline Create logical in	nstance	
1	e Empname	string	1	1	Representation Properties (×)= V	ariables	
:	e Address		1	1		10 00000000000000000000000000000000000	1
	sequence		1	1	TwoCharsText (Type)		(
1	e StreetName	string	1	1			_
	e City	string	1	1	<type filter="" text=""></type>	📑 🔜 🗙 🔆 🖽	
	e ZipCode	string	1	1	Property	Value	2
1	e Tel	<string></string>	1	1	Data Format Reference	<pre><default format=""></default></pre>	
1	e Salary	decimal	1	1	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
Add a Local Element					Byte Order	昂 <dynamically set=""></dynamically>	
					Ignore Case	<mark>暑 no</mark>	
•Simple Ty	pes 🔄 🕱				Fill Byte	暑 0	
		more simple el	ements.		Fill Byte	昂 0 short	
	pes ଟ 🗱 defines the allowed values for one or	more simple el	ements.				
Simple Type		more simple el	ements.		⊡ Content	short	
A simple type	defines the allowed values for one or Base Type	more simple el	ements.		Content Representation	short Patext	
A simple type	defines the allowed values for one or	more simple el	ements.		Content Representation Length Kind	short Patext	
Name	defines the allowed values for one or Base Type	more simple el	ements.		Content Representation Length Kind Text Content	short 2. text 2. delimited	
A simple type Name Data Form	defines the allowed values for one or Base Type TwoCharsText short hats (1 format)	more simple el	ements.		Content Representation Length Kind Text Content Number Representation	short	
A simple type Name Data Form	defines the allowed values for one or Base Type TwoCharsText short	more simple el	ements.		Content Representation Length Kind Text Content Number Representation Number Justification	short	
A simple type Name Data Form A data format	defines the allowed values for one or Base Type TwoCharsText short hats (1 format)	more simple el	ements.		Content Representation Length Kind Text Content Number Representation Number Justification Number Pad Character	short	

9. First, the Content Representation has been set to "text". This is the correct value for this scenario.

Second, the "Length kind" is set to "delimited". Change this to "explicit".

The editor will then provide two further properties. Set Length to 2, and leave Length Units as "characters".

Content	short
Representation	暑 text
🖃 Length Kind	explicit
Length	2
Length Units	🖥 characters

Note that changing lengthKind from 'delimited' to 'explicit' does not necessarily mean there is no delimiter present, it means that the parser does not scan for the delimiter to establish the length.

10. Finally, when the number representation is "text", the "Number Pattern" must have a defined value (it will be set to <unset>).

In the "number pattern" field, type '00' (without the quotation marks), and click return. (You can also use the wizard button for more complex patterns, but not required in this case).

All other text number properties of the simple type can be left as they are.
--

Content	short
Representation	扂 text
🖃 Length Kind	explicit
Length	2
Length Units	🛃 characters
Text Content	
Number Representation	🛃 standard
Number Base	昂 10
Number Check Policy	<mark>恳 lax</mark>
Number Pattern	00
Grouping Separator	器 ,
Decimal Separator	

11. You have now defined the Simple Type (TwoCharsText) that we will reference from the elements in the main model.

Save the model (Ctrl-S).

12. Now switch to the CompanyAddressChar message.

The three elements under the Address element need to be changed to use the TwoCharsText simple type that you just defined.

est Parse Model 1	Fest Serialize Model Hide pro	perties Show t	20 A2012	ctions Focus on selected Show quick o	utline Create logical instance
Messages message is a globa	I element that models an enti	re document of a	jata.	Representation Properties	Asserts and Discriminators 🕅 🕬= Variab
Name		Туре	Min Occurs	<type filter="" text=""></type>	
🖃 🖻 Comp	anyAddressChar			Property	Value
🖃 🚥 se	quence		1	Comment S	
: e	CompanyName	string	1	🖃 General	
Employee			1	Data Format Reference	<pre><default format=""></default></pre>
E	see sequence		1	Encoding (code page)	🛃 <dynamically set=""></dynamically>
1	e EmpNo	integer	1	Byte Order	🛃 <dynamically set=""></dynamically>
1	e Dept	integer	1	Ignore Case	昂 no
1	e Empname	string	1	Fill Byte	見の
1	🖃 🖻 Address		1	Content	string
	🖃 🚥 sequence		1	Representation	🛃 text
1	e StreetName	string	1	Length Kind	🛃 delimited
:	e City	string	1	- → Nillable S	🛃 false
1	e ZipCode	string	1	Default Value 😒	<unset></unset>
-	e Tel	<string></string>	1	Fixed Value 😒	<unset></unset>
	e Salary	decimal	1	🖃 Text Content	見 left

13. Highlight the StreetName element, and make the following changes to the Representation Properties of this element (Content section).

Representation = text Length Kind = prefixed

When you set the Length Kind to Prefixed, the editor provides further properties which allow you to set additional value. Use the drop-down value to select the following values:

Length Units = characters Prefix Length Type = TwoCharsText Prefix Includes Prefix Length = no.

e Address	1 1		1	Ignore Case 🔤 no
	1 1		1	Fill Byte 📃 0
e StreetName	string	1	1	Content string
e City	string	1	1	Representation 🚪 text
e ZipCode	string	1	1	Length Kind prefixed
e Tel	<string></string>	1	1	Length Units 📱 characters
e Salary	decimal	1	1	Prefix Length Type TwoCharsText
Add a Local Element				Prefix Includes Prefix Length no
				The

14. Make the same changes to the City and ZipCode elements.

Save the model.

Now you are done, and ready to test the new model!

15. Click the Test Parse Model button.

Select "Content from a data file", and click Browse. Select the Company.Prefix.Char.txt file, and click OK, and then OK again and then confirm the switch to the DFDL perspective.

-	integration Develop	🌐 Test Parse	Model	
File		Select mes Message r Parser Inp Select con C Contei	File Selection	
₽ 82 82 84 84 84 84 84 84 84 84 84 84 84 84 84	▼Schema Namespace <nl ▶Schema Refe A schema file in tt</nl 	© Contei Input file r Specify runt Runtime e Provide ru	library.descriptor	De dynamically set. More
	A message is a globa	Encoding (Floating pi Byte order Runtime v Validat Restore Defr	Select an input file from the file system Browse OK	

16. Success!

(Or perhaps not)

est Parse I	B Model Test Serialize Model Show	properties Sho	all sections	Focus on selecte	남 d Show quick outlin	e Create logical instanc	•	Data source: <from 'dfdi<br="">Message: CompanyAddres</from>		
essages essage is a	हा 😰 कि 🦊 🕱 a global element that models an er	🖪 🗟	data.					Tree View XML View		
	8							Name	Туре	Value
Name		Type	Min Occurs	Max Occurs	Default Value	Sample Value		CompanyAddressChar		
= e	CompanyAddressChar							CompanyName	xs:string	My Co
E	sequence		1	1				± Employee		
	e CompanyName	string	1	1		a .				
	e Employee		1	unbounded						
	🖃 🚥 sequence		1	1						
	e EmpNo	integer	1	1		1				
	e Dept	integer	1	1		1				
	e Empname	string	1	1	body value3	body value3				
You are ad	errors were encountered during p lvised to read the DFDL Trace to fi ng Errors E: Separator ',' not found at offse sedDataRegion[SimpleContent, st	nd out the root ca t '102' for sequer	ice within elemer	nt '/CompanyAd	dressChar[1]/Employ	ee[1]/Address[1]'.				del::sequen
	ceived during parsing are highlight									
	the trace captured while running th					and the second sec				
To view t	the partial logical instance that was				-					
	menu on the view toolbar provide	s options to contr	or now the data	is displayed in the	he view. Click the arro	ow icon on the toolbar or	nere	to open the menu.		
The view	display this message again									

What did you do wrong?

Close the yellow parser output message.

See if you can work out what went wrong by using the Test Parser output messages, and the highlighting in the Test - Parse window. You may also find it useful to take a look at the parse trace file, easily accessed by clicking on the link in the Test - Parse window.

Ra: Naviga pr 🖾 Problems 🕞 DFDL Test - Parse 🐼 🕞 DFDL Test - Serialize 🕞 DI DL Test - Trace
DFDL tes - Parse: Runs the DFDL parser with the provided physical input data and selected in message, and updates the logical instance view with the result of the parse.
Status: Parsing completed with processing errors: Thu Apr 05 04:40:37 CDT 2012
Data: //MessageModellingPrefixLib/Company.Prefix.Char.txt 💌 Browse Message: CompanyAddressChar (/MessageModellingPrefixLib
Parsed Input Characters 1 Company [compName=My Company 2 Employee (empNum=111111] dept=500 empName=Alice Wong Addr: 158200 Warden Ave 4 "Markham, Ont"07L3G 1H7 3 Employee (empNum=222222 dept=500 empName=James May Addr:1523 The Cuttings07Chatham07CH2 2PR tel=208-203- 4 Employee (empNum=33333] dept=310 empName=Richard Hammond Addr:1716 Great Windmill06London06W2 3RJ tel=21 5 Employee (empNum=44444 dept=230 empName=Jaremy Clarkeson Addr:22"Rose Cottage, Pea Dr"10Gloucester08GLU 6 Employee (empNum=555555 dept=650 empName=Humphrey Littleton Addr:17416 Regent Street06London07NW1 1QT te 7] 8

If you can't work this out, proceed to the next page . . .

Page intentionally left blank to give you time to work out what went wrong . . .

17. Well, the clues are fairly clear in fact. The parse failure message says that a separator is missing for a sequence within the Address element. Now the changes that you have made in this lab have changed the parsing of the elements under Address from using a separator, to using the prefix length. So, why is the model still expecting a separator (and not finding one in the test data).

Come to think of it, you didn't actually make a change to the separator definition, did you?

Switch back to the Integration Development perspective, and take a look at the Address sequence field in the editor. You will see that the separator for the sequence element is still set to ',' (ie. a comma). So, the model is expecting these fields to be delimited by a comma, and of course our data does not match this model.

CompanyAddressChar.xsd 🛛 🗖 é							
Test Parse Model Test Serialize Model Hide pr	operties Show	all section	Focus on select	ted Show quick outline Create logical in	nstance		
Messages 🕢 🖟 🖉 🕆 💥 message is a global element that models an entir	e document of o	lata.	<u>^</u>	Representation Properties (*)= V	/ariables 📔 Asserts and Discriminators	(?)	
Name	Туре		curs Max Occ	<type filter="" text=""></type>	· · · · · · · · · · · · · · · · · · ·		
e CompanyAddressChar				Property	Value	(?)	
		1	1	🖃 General			
e CompanyName	string	1	1	Data Format Reference	<pre><default format=""></default></pre>		
e Employee		1	unbound	Encoding (code page)	R ≤dynamically set>		
		1	1	Byte Order	器 <dynamically set=""></dynamically>		
e EmpNo	integer	1	1	Ignore Case	昂 no		
e Dept	integer	1	1	Fill Byte	昂 0		
e Empname	string	1	1	Content			
E Address	Jung	1	1	Initiated Content	昂 no		
	-	1	1	Sequence Kind	界 ordered		
StreetName	string	1	1	Occurrences			
		1		Min Occurs S	昂 1	_	
e City	string		1	Max Occurs S	界 1		
e ZipCode	string	1	1	Alignment			
e Tel	<string></string>	1	1	Delimiters			
e Salary	decimal	1	1		昆,		
Add a Local Element				Initiation	27 ALO HIGGOLY		

18. Change the separator to "no separator" (use the delete key do not set the separator to a blank character).

🛨 Alignment	
Delimiters	
	<no separator=""></no>
Initiator	뿸 <no initiator=""></no>

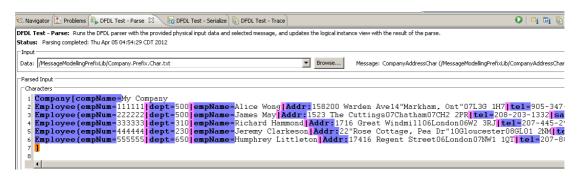
Save the model.

19. Now retest the model. This time... success !

😪 Navigator 📳 Problems 🕒 DFDL Test - Parse 🛛 🔓 DFDL Test - Serialize) 🕼 DFD	
	③ Parsing completed successfully.
DFDL Test - Parse: Runs the DFDL parser with the provided physical input data and selected	
Status: Parsing completed: Thu Apr 05 04:54:29 CDT 2012	
Input	Tips:
Data: //MessageModellingPrefixLib/Company.Prefix.Char.txt	 Selecting an element in the DFDL editor will cause the parsed input to focus only on data pertaining to the selected
	• The view menu on the view toolbar provides options to control how the data is displayed in the view. Click the arrow
Parsed Input	• To view the logical instance that was created by the DFDL parser, click the Open DFDL Logical Instance View toolba
Characters	• To view the trace captured while running the DFDL parser, click the Open DFDL Trace View toolbar button, or click t
1 Company [compName=My Company	
2 Employee (empNum=111111 dept=500 empName=Alia	
3 Employee (empNum=222222 dept=500 empName=Jame	
	hard Hammond Addr: 1716 Great WindmillU6LondonU6W2 3RJ tel=207-4
	emy Clarkeson Addr: 22"Rose Cottage, Pea Dr"10Gloucester08GL01 2
B Emproyee (empNum=ppppbbb dept=b50 empName=Hum)	phrey Littleton <mark> Addr:</mark> 17416 Regent StreetO6LondonO7NW1 1QT <mark> tel=</mark> 2
0	

20. Close the yellow completion pop-up.

The parsed data will be seen in the Test – Parse window.



21. In the Logical Instance window, expand the Tree View, and expand the Address element in one or two of the employee elements. You will see that the message has been fully parsed. The prefix length does not show in the Tree View (it is not treated as part of the message data), although it is displayed in the Test - Parse window.

🗏 i DFDL Test - Logical Instance 🕱 🛛 🖥 🗟 🖓							
Data source: <from 'dfdl="" -="" parse'="" test="" view=""></from>							
Message: CompanyAddress	Char (/workspace	es/DFDL/MessageModelling	PrefixLib/CompanyA				
Tree View XML View							
Name	Туре	Value					
🗖 CompanyAddressChar							
CompanyName	xs:string	My Company					
Employee							
Employee							
EmpNo	xs:integer	222222					
Dept	xs:integer	500					
Empname	xs:string	James May					
 Address 							
StreetName	xs:string	23 The Cuttings					
City	xs:string	Chatham					
ZipCode	xs:string	CH2 2PR					
Tel	xs:string	208-203-1332					
Salary	xs:decimal	189599.95					
🖃 Employee							
EmpNo	xs:integer	333333					
Dept	xs:integer	310					
Empname	xs:string	Richard Hammond					
Address							
StreetName	xs:string	16 Great Windmill					
City	xs:string	London					
ZipCode	xs:string	W2 3RJ					
Tel	xs:string	207-445-2955	•				

This concludes the Prefix Length Character scenario.

4. Create the Prefix Length Binary scenario

1. Close the Test Parse perspective, and close the CompanyAddressChar message model.

Open and expand the CompanyAddressBin.xsd message model.

t Parse Mo	del Test Serialize Model Hide propert	ties Show all s	sections Focus	on selected Sł	ta E now quick outline Create logical instance	6	
Message	s 🛛 🗐 🔊 🕆 🗱	E b		_	Representation Properties		
	is a global element that models an entire		data.		CompanyAddressBin (Element)		
Name	~	Туре	Min Occurs	Max Occurs	Þ	🏩 🙀 🗱 🗱	
	e CompanyAddressBin				Property	Value	?
	🖃 🚥 sequence		1	1	Comment S		
1	e CompanyName	string	1	1	General		
1	🖃 e Employee		1	unbounded	Data Format Reference	<pre><default format=""></default></pre>	
	🖃 🚥 sequence		1	1	Encoding (code page)	器 <dynamically set=""></dynamically>	
1	e EmpNo	integer	1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
1	e Dept	integer	1	1	Ignore Case	昂 no	
1	e Empname	string	1	1	Fill Byte	晃 0	
	G Address		1	1	Content		
	sequence		1	1	Length Kind	🛃 delimited	
1	e StreetName	string	1	1	Occurrences		
	e City	string	1	1	Min Occurs S	昂 1	
1	e ZipCode	string	1	1	Max Occurs S	累 1	
	e Tel	<string></string>	1	1	Alignment		
	e Salary	decimal	1	1	Delimiters		
Add a L	.ocal Element				Initiator	Company[
]%CR;%LF;	

2. Highlight the Address element. You will see that the separator has been set to ','; this means that all fields in the Address element are separated by commas.

This is the part of the model that we will change.

) CompanyAddressBin.xsd 🛙 🖓								
Est Parse Model Test Serialize Model Hide properties Show all sections Focus on selected Show quick outline Create logical instance								
► Messages Image: Im	document of d	data.		Representation Properties Sequence	Variables] E Asserts and Discriminators]			
Name	Туре	Min Occurs	Max Occurs	N	詩 昆 🗶 🍇 🗉 🖬 🖶			
🖃 🖻 CompanyAddressBin				Property	Value (?)			
🖃 🚥 sequence		1	1	General				
E CompanyName	string	1	1	Data Format Reference	<default format=""></default>			
: e Employee		1	unbounded	Encoding (code page)	R <dynamically set=""></dynamically>			
□ ••• sequence		1	1	Byte Order	🛃 <dynamically set=""></dynamically>			
e EmpNo	integer	1	1	Ignore Case	昇 no			
e Dept	integer	1	1	Fill Byte	昇 0			
e Empname	string	1	1	Content				
E e Address		1	1	Initiated Content	暑 no			
		1	1	Sequence Kind	🛃 ordered			
e StreetName	string	1	1	Occurrences				
		1	1	Min Occurs S	累 1			
e City	string			Max Occurs S	昂 1			
E ZipCode	string	1	1	Alignment				
e Tel	<string></string>	1	1	Delimiters	1			
e Salary	decimal	1	1	Separator	昂, く …			
Add a Local Element				Initiator	🛃 <no initiator=""> 📃</no>			
				Terminator	🛃 <no terminator=""> 🛛</no>			
				Output New Line	晃 %CR;%LF;			

3. You will change the elements within the Address element to be identified and parsed by using length prefixes.

In this scenario, each of the elements under the Address element will have a prefix of length 2 bytes. The prefix will indicate the length of the element, and the value of the prefix will be a two's complement binary integer. In this case, the value contained in the length prefix will include the length of the prefix itself, unlike the character scenario.

The Address element may look like this:

Addr: ¤8200 Warden Ave "Markham, Ont" L3G 1H7

4. To define this type of model, you first need to define a Simple Type. This is used to define the physical characteristics of the prefix length. An element which has a prefix length then simply refers to the simple type.

Click "Show all sections" on the main editor line.

0 Co	mpanyAddre	ssBin.xsd 🛛								
Test F	Parse Model	Test Serialize	Model	Hide properties	A Show all s	ections	Focus			
I —	▪Messages 🛛 🙀 🛊 🌵 🐳 🛛 🖡 🖬 A message is a global element that models an entire document of data.									
	Name				Туре	Min	Occurs			
	= e (
		sequence		1						
	1	e Compan	string	1						
	1	🖃 🖻 Employe	e			1				
		_								

5. In the main editor pane, expand Simple Types, and then click the "Add Simple Type" button.



6. In the dialogue window, set Name = TwoBytesBin (you can define your own descriptive name for this type), and set "Inherit from" to short.

Click OK.

🕀 Add Sim	ple Type		<u>_ ×</u>
Name:	TwoBytesBin		
Inherit from:	short		▼
		ОК	Cancel

7. To make the editor clearer, click "Hide empty sections".

E Parse Model Test Serialize Model Hide proper	ties Hide empt	ty sections Fo	ocus
Name	Туре	Min Occurs	Ma
e CompanyAddressBin			
sequence		1	1
E CompanyName	string	1	1
Employee		1	un
🖃 🚥 sequence		1	1
e EmpNo	integer	1	1
e Dept	integer	1	1
e Empname	string	1	1
E e Address		1	1
🖃 🚥 sequence		1	1
e StreetName	string	1	1
e City	string	1	1
e ZipCode	string	1	1
e Tel	<string></string>	1	1
e Salary	decimal	1	1

8. Highlight the new Simple Type. You will see that various properties have been set for this new type, shown in the Representation Properties in the right hand pane. Some of these properties must be changed to reflect the nature of our prefix length values.

0 *CompanyAddressBin.xsd 🛛						
Test Parse Model F	ide properties Hide emp	ty sections	Focus on select	ed Show quick outline Create logical inst	tance	
e Empname	string	1	1	Representation Properties)= Variables	
E e Address		1	1			
🚍 🚥 sequen	ce	1	1	TwoBytesBin (Type)		0
: e Stre	etName string	1	1		Tenganet Courses 1	
: e City	string	1	1	<type filter="" text=""></type>	⇒ 昂	🗙 💥 🖪 🖻 🛤
E ZipC	Code string	1	1	Property	Value	()
E Tel	<string></string>	1	1	Comment S		
e Salary	decimal	1	1	General		
Add a Local Element				Data Format Reference	<pre><default format=""></default></pre>	
				Encoding (code page)	<pre>set></pre>	
Global Elements (0 elements)				Byte Order	🛃 <dynamically set=""></dynamically>	
A global element represents a named in	tance of a complex or sim	ple type.		Ignore Case	昂 no	
		pre type:		Fill Byte	昂 0	
Complex Types (0 complex type	s)			Content	short	
A complex type defines the elements an		structure		Representation	暑 text	
A complex type dennes are elements an	a groups and represent a	Su actare.		Length Kind	🛃 delimited	
▼Simple Types 🛛 📰 🕱				Text Content		
				Number Representation	🛃 standard	
A simple type defines the allowed values	s for one or more simple ei	ements.		Number Justification	🛃 right	
Name Base T	/De			Number Pad Character	昂 %SP;	
	/Pc			Pad Kind	🛃 none	
TwoBytesBin short				Trim Kind	🔒 padChar	
				Escano Schama Dafaranca	E rocConEioldaEmtrDocor	edEaconoSchor

9. In the Content section, Representation has been set to "text". Change this to "binary".

Second, the "Length kind" has been set to "delimited". Change this to "explicit".

The editor will then provide two further properties. Set Length to 2, and set Length Units to "bytes".

Content	short	
Representation	binary	
🖃 Length Kind	explicit	
Length	2	
Length Units	bytes	

10. Finally, when the number representation is "binary", the "Binary Number Representation" must have a defined value. Set this to "binary". This means that the value is a "two's complement" integer.

Set the Binary Number Check Policy to "lax". It is required to set this property for binary elements, but you can also set it to "Strict" in this example.

Content	short		
Representation	binary		
Length Kind	explicit		
Length	2		
Length Units	bytes		
Binary Content			
Number Check Policy	lax		
Number Representation	binary		

11. You have now defined the Simple Type (TwoBytesBin) that we will reference from the elements in the main model.

Save the model (Ctrl-S).

12. Now switch to the CompanyAddressBin model.

The three elements under the Address element need to be changed to use the TwoBytesBin simple type element that you just defined.

13. Highlight the StreetName element, and make the following changes to the Representation Properties of this element (Content section).

Representation = text Length Kind = prefixed

When Length Kind is set to "prefixed", further properties should be set as follows: Length Units = bytes Prefix Length Type = TwoBytesBin

Prefix Includes Prefix Length = yes (this means the length value will include the length of the prefix itself)

1	🖃 e Address		1	1	E Content string	
	sequence		1	1	Representation 🛛 🔤 text	
1	e StreetName	string	1	1	Length Kind prefixed	
1	e City	string	1	1	Length Units bytes	
1	e ZipCode	string	1	1	Prefix Length Type TwoBytesBin	
1	e Tel	<string></string>	1	1	Prefix Includes Prefix Lengt yes	
1	e Salary	decimal	1	1	Nillable S A false	

Make the same changes to the City and ZipCode elements.

14. As in the first scenario, you now need to remove the separator from the Address sequence.

You will see that the separator for the sequence element is still set to ',' (ie. a comma).

lessages	2 🖉 û 🖞 🗶	E E		4	Representation Properties	Variables 🗄 Asserts and Discriminato	rs
iessage is a global e	lement that models an entire	e document of (lata.		sequence		
Name		Туре	Min Occurs	Max Occurs	<type filter="" text=""></type>	🔆 🕺 🙀	E E
😑 🖻 Compan	yAddressBin				Property	Value	(?)
🖃 🚥 sequ	ence		1	1	🖃 General		
eC	CompanyName	string	1	1	Data Format Reference	<default format=""></default>	
: 😑 e E	imployee		1	unbounded	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
	•• sequence		1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
	e EmpNo	integer	1	1	Ignore Case		
	e Dept	integer	1	1	Fill Byte	晃 0	
	e Empname	string	1	1	Content		
1	= e Address		1	1	Initiated Content	昇 no	
	🖃 🚥 sequence		1	1	Sequence Kind	🛃 ordered	
	e StreetName	string	1	1	Occurrences		
	e City	string	1	1	Min Occurs S	暑 1	
	e ZipCode	string	1	1	Max Occurs S	冕 1	
	e Tel	<string></string>	1	1	Alignment		
1	e Salary	decimal	1	1	Delimiters		
Add a Local Elemen					Separator	昂,	

15. Change the separator to "no separator" (use the delete key do not set the separator to a blank character).

Alignment	
Delimiters	
	<no separator=""></no>
Initiator	暑 <no initiator=""></no>
	—

Save the model.

16. Click the Test Parse Model button.

Select "Content from a data file", and click Browse. Select the Company.Prefix.Bin.PrefixIncludingLength.txt file, and click OK, and then OK again.

Do not use the file Company.Prefix.Bin.txt that is a test file with data where the length prefix does not contain the length of the prefix itself..... that model is left as an exercise for the reader.

۲	Integration Develop	🌐 Test Parse	Model	_ _ _ ×
File	Edit Navigate Se			
2	- 🗌 🖻 🖆	Select mes	File Selection	
8 13 13 14 14 14 14 14 14 14 14 14 14 14 14 14	CompanyAddress Test Parse Model 1 •Messages	Parser Inc Select con C Conter C Conter	MessageModellingPrefixLib IBMdefined Company.Prefix.Bin.PrefixIncludingLength.txt Company.Prefix.Bin.txt Company.Prefix.Char.txt Ibrary.descriptor	
84 84 84 84	A message is a globa	Input file r		Browse
M	Name Comp Name Name Name Second	Specify runt Runtime e Provide ru Encoding (Floating pr		pe dynamically set. <u>More</u>
		Byte order Runtime v	Select an input file from the file system Browse	
		Restore Defa	OK Cance	

17. Close the yellow completion pop-up.

😵 Navigator 🔝 Problems 🕒 DFDL Test - Parse 🛛 🔓 DFDL Test - Serialize 🗈 DFD	Parsing completed successfully.			
DFDL Test - Parse: Runs the DFDL parser with the provided physical input data and selected	• · · · · · · · · · · · · · · · · · · ·			
Status: Parsing completed: Thu Apr 05 04:54:29 CDT 2012				
Input	Tips:			
Data: //MessageModellingPrefixLib/Company.Prefix.Char.txt	• Selecting an element in the DFDL editor will cause the parsed input to focus only on data pertaining to the selected			
	• The view menu on the view toolbar provides options to control how the data is displayed in the view. Click the arrow			
Parsed Input	• To view the logical instance that was created by the DFDL parser, click the Open DFDL Logical Instance View toolba			
Characters	• To view the trace captured while running the DFDL parser, click the Open DFDL Trace View toolbar button, or click [
1 Company[compName=My Company				
2 Employee (empNum=111111 dept=500 empName=Alic				
3 Employee (empNum=222222 dept=500 empName=Jame				
4 Employee (empNum=333333 dept=310 empName=Richard Hammond Addr:1/16 Great WindmillUbLondonU6W2 3RJ tel=207-4				
	emy Clarkeson <mark> Addr:</mark> 22"Rose Cottage, Pea Dr"10Gloucester08GL01_21			
6 Employee (empNum=5555555 dept=650 empName=Hump	phrey Littleton Addr: 17416 Regent Street06London07NW1 1QT tel= 2			
7				
8				

The parsed data will be seen in the Test – Parse window.

😪 Navigator 📳 Problems 🖺 DFDL Test - Parse 🙁 🔪 🔄 DFDL Test - Serialize) 🕼 DFDL Test - Trace	O 📑 🖬 🕼 📾 🔗
DFDL Test - Parse: Runs the DFDL parser with the provided physical input data and selected message, and updates the logical instance view with the result of the parse.	
Status: Parsing completed: Thu Apr 05 05:49:20 CDT 2012	
_ Input	
Data: //MessageModellingPrefixLib/Company.Prefix.Bin.bxt 💌 Browse Message: CompanyAddressBin (/MessageModellingPrefixLib/Com	mpanyAddressBin.×sd)
Parsed Input	
Characters	
<pre>1 Company[compName=My Company 2 Employee(empNum=11111] dept=500] empName=Alice Wong Addr:W8200 Warden Avell"Markham, Ont"WL3G 1H7 tel 3 Employee(empNum=22222) dept=500] empName=James May[Addr:W33 The CuttingsIIChathamIICH2 2PR] tel=208-2 4 Employee(empNum=333333] dept=310] empName=Richard Hammond Addr:W16 Great WindmillIILondonIIW2 3RJ tel 5 Employee(empNum=44444] dept=230] empName=Jaremy Clarkeson Addr:W16 Great WindmillIILondonIIW2 3RJ tel 6 GloucestrIIGL0 12NM tel=743-123-4567] sal-75599.95 7 Employee(empNum=555555] dept=650] empName=Humphrey Littleton Addr:W4II Regent StreetILondonIINW1 1QT</pre>	03-1332 <mark> sal=</mark> 189599. =207-445-2955 <mark> sal=</mark> 5

18. In the Logical Instance window, expand the Tree View, and expand the Address element in one or two of the employee elements. You will see that the message has been fully parsed. The prefix length does not show in the Tree View (it is not treated as part of the message data), although it is displayed in the Test - Parse window.

i DFDL Test - Logical Instanc	ie 🛛					
ata source: <from 'dfdl'<="" th=""><th>Test - Parse' view</th><th>></th></from>	Test - Parse' view	>				
Accesses CompanyAddross	Pin (Juarkanacaa	/mm5/MessageModellingPrefix				
CompanyAddress	bin (workspaces	mmo/messagemodellingFrenz				
Tree View XML View						
Name	Туре	Value				
CompanyAddressBin						
CompanyName	xs:string	My Company				
 Employee 						
 Employee 						
EmpNo	xs:integer	222222				
Dept	xs:integer	500				
Empname	xs:string	James May				
Address						
StreetName	xs:string	23 The Cuttings				
City	xs:string	Chatham				
ZipCode	xs:string	CH2 2PR				
Tel	xs:string	208-203-1332				
Salary	xs:decimal	189599.95				
 Employee 						
EmpNo	xs:integer	333333				
Dept	xs:integer	310				
Empname	xs:string	Richard Hammond				
 Address 						
StreetName	xs:string	16 Great Windmill				
City	xs:string	London				
ZipCode	xs:string	W2 3RJ				
Tel	xs:string	207-445-2955				
Salary	xs:decimal	599.95				

END OF LAB GUIDE