

IBM Integration Bus

Message Modeling with DFDL

Lab 5 Using DFDL length prefixes

June 2015 Hands-on lab built at product Version 10.0.0.0 16L95

1.	INTE	RODUCTION	3
	1.1 1.2	LAB PREPARATION LAB SCENARIO	
2.	IMP	ORT THE BASE MODEL	1
3.	CRE	EATE THE PREFIX LENGTH CHARACTER SCENARIO	3
4.	CRE	EATE THE PREFIX LENGTH BINARY SCENARIO20)
E١	ID OF	LAB GUIDE24	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

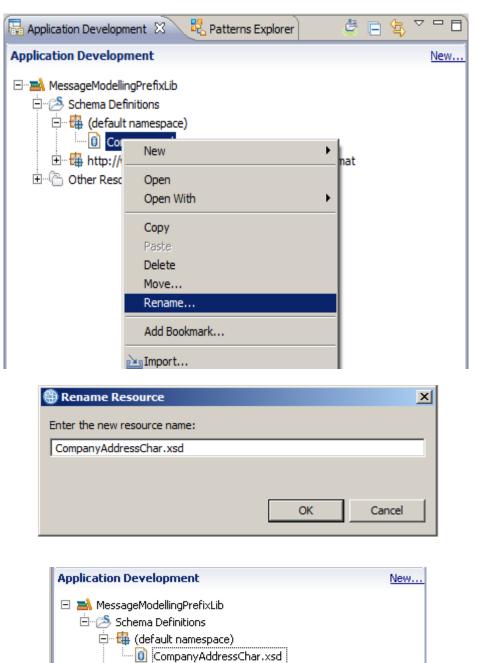
You are going to create two message models. One will use a length prefix in character form, and one will use a length prefix in binary form. Both length prefixes will be two bytes.

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.

Integration Development - File Edit Navigate Search Pro	IBM Integration Toolkit - C:\workspaces\mm5	_ [] ×
		Quick Access
🔚 Application Dev 🛛 💐 Pat		
		1
Application Development	Import Project Interchange Contents	
New Application New Integration Service New REST API	Import Projects Import Projects from a zip file.	
New Library	From zip file: C:\student10\WessageModeling\prefixes\PrefixLabSt	
	Project location root: C:\workspaces\mm5 Browse	
응 Int 23 % Int 또 Dat	MessageModellingPrefixLib	
Graint 25 Baint to Date Gaint to Date	Select All Select Referenced	mentlog 🛃 🛱 🛱 🖓 🗔 🗸 🗖 🗖
	Cancel	
Ľ		

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



🗄 📲 http://www.ibm.com/dfdl/RecordSeparatedFieldFormat

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	🖉 🖻 🤹 🗸 🗖 🗖
Application Development	<u>New</u>
MessageModellingPrefixLib Schema Definitions Gradie (default namespace) CompanyAddressBin.xsd CompanyAddressChar.xsd Http://www.ibm.com/dfdl/RecordSeparatedF CompanyAddress	ieldFormat

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

OmpanyAddressBin.xsd ⋈										
Test Parse Model Test Serialize Mode	l Hide prop	erties Show	all sections	Focu						
Messages 🛛 🙀 🕆 🦑 💥 📙 🖬 message is a global element that models an entire document of data.										
Nar <mark>ne</mark>	Туре	Min Occurs	Max Occurs							
e 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.

CompanyAddressChar.xsd											
Test Parse Model	E Test Serialize Model	Hide prope	erties Show	all sections	Focu						
Messages 🛛 💭 🕼 🗘 🦑 💥 📙 🖬 message is a global element that models an entire document of data.											
Name		Туре	Min Occurs	Max Occurs							
e Compa	anyAddressChar										
🖃 🚥 Se	quence		1	1							
: e	CompanyName	string	1	1							
: ± e	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.

) CompanyAddressChar.xsd 🛙 🗁 🗗									
E Test Parse Model	E0 III Test Serialize Model Hide propertie	es Show all s	ections Focus	on selec	ted Show quick outline Create logical insta	nce			
▼Messages		E . E .		<u> </u>	Representation Properties				
	lobal element that models an entire		ata.		CompanyAddressChar (Element)		?		
Name		Туре	Min Occurs	Max	Þ	📑 🛃 🗶 💥 🗉	₽ ♣		
- e 9	ompanyAddressChar				Property	Value	?		
	•• sequence		1	1	Comment S				
1	e CompanyName	string	1	1	General				
8 8	= e Employee		1	unbo	Data Format Reference	<default format=""></default>			
	🖃 🚥 sequence		1	1	Encoding (code page)	🛃 <dynamically set=""></dynamically>			
1	e EmpNo	integer	1	1	Byte Order	🛃 <dynamically set=""></dynamically>			
1	e Dept	integer	1	1	Ignore Case	뤔 no			
1	e Empname	string	1	1	Fill Byte	昂 0			
1	e Address		1	1	Content				
	sequence		1	1	Length Kind	暑 delimited			
1	e StreetName	string	1	1	Occurrences				
	e City	string	1	1	Min Occurs S	暑 1			
1.1	e ZipCode	string	1	1	Max Occurs S	晃 1			
	e Tel	<string></string>	1	1	Alignment				
			-	1	Delimiters				
i Add a Level	e Salary	decimal	1	1	Initiator	Company[
Add a Local	Element				Terminator]%CR;%LF;			
					Empty Value Delimiter Policy	界 initiator			
					Output New Line	暑 %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.

CompanyAddressChar.xsd						- 6				
Es Es Constant de la Constant de la Constant de Consta										
▼Schema 🖉 🗿 🗿 💭 💭) (<u>↓ 6</u> ▲	Representation Properties (X)= Varia	ables 🗄 Asserts and Discriminate	ors				
				sequence		0				
Namespace <null namespace=""></null>			_							
Schema References (0 includes, 1 impor	t)		_	<type filter="" text=""></type>	추 🖪	🗙 🔆 🗎 🖽 🖃 🖶				
A schema file in the same namespace uses an ind	ude. A schema	a file in a diffe	erent 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 d	lata.		Encoding (code page)	🛃 <dynamically set=""></dynamically>					
				Byte Order	🛃 <dynamically set=""></dynamically>					
Name	Туре	Min Occu	rs Max	Ignore Case	<mark>昂 no</mark>					
CompanyAddressChar	1764	r in r o cco		Fill Byte	晃 0					
				Content						
e e sequence		1	1	Initiated Content	🛃 no					
e CompanyName	string	1	1	Sequence Kind	🛃 ordered					
Employee		1	unbo	Occurrences						
sequence		1	1 -	Min Occurs S	晃 1					
e 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	🛃 <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.

OmpanyAddressChar.xsd ∑								
Test Parse Model Test Serialize Model Hide properti	A Show all sections	incus on selec	selected Show quick outline Create logical instance					
	J 🛃 🗄 🖬	E. E	Representation Properties 🛛 🕬= Varia	bles 🖹 Asserts and Discriminators				
Namespace <null namespace=""></null>			sequence		0			
Humespace (Humespace)			I					
▶ Schema References (0 includes, 1 impor	t)		<type filter="" text=""></type>	🗦 🛃 🗙 💥	🕀 🖻 🖶			
A schema file in the same namespace uses an ind	ude. A schema file in a di	fferent na	Property	Value	3			
			General					
-Messages 🛛 🔄 🕼 🕆 💥	6 6		Data Format Reference	<default format=""></default>				
A message is a global element that models an entire	document of data.		Encoding (code page)	🛃 <dynamically set=""></dynamically>				
N			Byte Order	dynamically set>				
Name	Type Min Oc	curs Max	Ignore Case	扂 no				
	1,72		Fill Byte	扂 0				
CompanyAddressChar			Content					
e e equence	1	1	Initiated Content	扂 no				
e CompanyName	string 1	1	Sequence Kind	🛃 ordered				
e Employee	1	unbo	Occurrences					
🖃 🚥 sequence	1	1	Min Occurs S	暑 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	🚪 <no initiator=""></no>				
E StreetName	string 1	1	Terminator	R <no terminator=""></no>				
: e City	string 1	1	Output New Line	界。%CR;%LF;				
: e ZipCode	string 1	1						
e Tel	<string> 1</string>	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.

			n ple Types ple type defines t	F ne allowe	X d values for one or more simple elements.
--	--	--	--	------------------	--

 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	ole Type		
Name:	TwoCharsText		
Inherit from:	short		▼
		ОК	Cancel

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

1) *CompanyAddressChar.xsd 🛛									
Test Parse Model Test Serialize Model Hide properti	s Hide emp	ty sections	For us on se	ected Show quick outline Create logical in	nstance				
E CompanyName	string	1	1 🔺	Representation Properties (X)= Varia	ables				
: e Employee		1	unbc				?		
🖃 🚥 sequence		1	1	TwoCharsText (Type)			U		
e EmpNo	integer	1	1	<type filter="" text=""></type>		n me l	÷		
: e Dept	integer	1	1		- Frank Franker		-8-		
Empname	string	1	1	Property	Value	?	_		
e Address		1	1	Comment S			-		
🖃 🚥 sequence		1	1	General					
e StreetName	string	1	1	Data Format Reference	<default format=""></default>		- 11		
: e City	string	1	1	Encoding (code page)	<pre></pre> dynamically set>		- 1		
e ZipCode	string	1	1	Byte Order	<pre></pre>		- 1		
e Tel	<string></string>	1	1	Ignore Case	揭 no		- 1		
e Salary	decimal	1	1	Fill Byte	晃 0 short				
Add a Local Element				Content					
				Representation	kext		- 1		
▶Global Elements (0 elements)				Length Kind	🛃 delimited				
A global element represents a named instance of a c	omplex or sim	nle tune		I ext Content I Number Representation	晃 standard		-		
A global element represents a named instance of a c	omplex or aim	pie type:		Number Representation	點 standard 晃 right		- 1		
Complex Types (0 complex types)				Number Justification			- 11		
A complex type defines the elements and groups that		about the second		Pad Kind	昂 %SP; 昂 none		- 11		
A complex type defines the elements and groups that	it represent a	istructure.		Trim Kind	昇 none 見 padChar		- 1		
▼Simple Types 🛛 🔚 🕱				Escape Scheme Reference	Patiential R recSepFieldsEmt:RecordEscapeScheme		- 11		
• · · ·				Alignment	The septeror second scapes of the second scapes of				
A simple type defines the allowed values for one or r	nore simple el	ements.		Delimiters			-		
Name Base Type				Initiator	💂 <no initiator=""></no>				
				Terminator	R <no terminator=""></no>		-		
TwoCharsText short			-	Empty Value Delimiter Policy	R initiator		-		
4			- P			-	-		

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 Model Test Serialize Model Hide prope		sections	Focus on sele	cted Show quick outline Create logical i	nstance	
e Empname	string	1	1	Representation Properties	Variables	
E e Address		1	1			
sequence		1	1	TwoCharsText (Type)		
e StreetName	string	1	1	<type filter="" text=""></type>	🔅 🛃 🗙 🔆 🖪	B
E ZipCode	string	1	1	Property	Value	?
E Tel	<string></string>	1	1	Data Format Reference	<pre><default format=""></default></pre>	
e Salary	decimal	1	1	Encoding (code page)	🖫 <dynamically set=""></dynamically>	
Add a Local Element				Byte Order	🛃 <dynamically set=""></dynamically>	
				Ignore Case	<mark>見 no</mark>	
Simple Types 🛛 📰 🗱				Fill Byte	昆 0	
simple type defines the allowed values for one of	or more simple el	ements.		Content	short	
				Representation	🔁 text	
Name Base Type				Length Kind	🔁 delimited	
= TwoCharsText short				Text Content		
				Number Representation	🖫 standard	
Data Formats (1 format)				Number Justification	界 right	
A data format is a container of DFDL properties.				Number Pad Character	昂 %SP;	
				Pad Kind	界 none	
Variables (4 variables)				Trim Kind	🛃 padChar	
A variables holds a value that can be used in DEDL expressions.				Escape Scheme Reference	R recSepFieldsFmt:RecordEscapeScheme	

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	昂 text
🖃 Length Kind	explicit
Length	2
Length Units	뿸 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).

Content	short
Representation	🛃 text
🖃 Length Kind	explicit
Length	2
Length Units	🛃 characters
Text Content	
Number Representation	🛃 standard
Number Base	暑 10
Number Check Policy	晃 lax
Number Pattern	00
Grouping Separator	嚣,
Decimal Separator	累.

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.

CompanyAddress		الف_		-2		-
e at Davies Maidal	Ei 🔲 🛄	→i →i			E	E
	Test Serialize Model Hide prope		asic Snow all se			
Messages		Ei Ei	1	Representation I	Properties 🔪 📘 Asser	ts and Discriminators 🕅 🗱 Variab
i message is a globi	al element that models an entire	document of c	ata.	StreetName (Elen	nent)	
Name		Туре	Min Occurs	<type filter="" text=""></type>		
🖃 🖻 Comp	panyAddressChar			Property		Value
= St	equence		1	Comment S		
	e CompanyName	string 1		🖃 General		
	Employee		1	Data Forma	at Reference	<default format=""></default>
6	= 🚥 sequence		1	Encoding (:ode page)	🛃 <dynamically set=""></dynamically>
1	e EmpNo	integer	1	Byte Order		🛃 <dynamically set=""></dynamically>
1	e Dept	integer 1		Ignore Cas	e	🛃 no
1	e Empname	string	1	Fill Byte		昇0
1	🖃 🖻 Address		1	🖃 Content		string
	🖃 🚥 sequence		1	Representa	ation	🛃 text
1	e StreetName	string	1	Length Kind	1	🛃 delimited
	e City	string	1	→ Nillable 🖻		🛃 false
-	e ZipCode	string	1	Default Val	ue S	<unset></unset>
1	e Tel	<string></string>	1	Fixed Value	s S	<unset></unset>
1	e Salary	decimal	1	🖃 Text Conten	:	
		accinar	•	String Justi	fication	🖳 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	Ignore Case	昂 no
🖃 🚥 sequence	1 1		1	Fill Byte	見 0
e StreetName	string	1	1	Content	string
e City	string	1	1	Representation	昂 text
e ZipCode	string	1	1	🖃 Length Kind	prefixed
e Tel	<string></string>	1	1	Length Units	뿸 characters
e Salary	decimal	1	1	Prefix Length Type	TwoCharsText
Add a Local Element				Prefix Includes Prefix Length	no
				T	- ·

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.

-	_	stion Develor	🌐 Test Parse	Model	
File	•	it Navigate Se	Message Select mes Message r Parser Inc Select con	File Selection Select an input file: Image: MessageModellingPrefixLib Image: Display input file Image: Display input file	
2010 B	* 9	st Parse Model 1 Schema Namespace <nu< td=""><td>C Conter C Conter Input file r</td><td>Company.Prefix.Bin.txt</td><td>Browse</td></nu<>	C Conter C Conter Input file r	Company.Prefix.Bin.txt	Browse
€ <u>-</u> ₩1		▶ Schema Refe A schema file in t	Specify runt Runtime e Provide ru		pe dynamically set. <u>More</u>
	II —	Messages nessage is a globa Name	Encoding (Floating p Byte order	Select an input file from the file system	<u> </u>
		- e Comp St :	Runtime v	OK Cancel	
		:			

(Or perhaps not)

est Parse Model	Test Serialize Model	Show properties	A Show all sections	Focus on selected	E Show quick outline	Create logical instance	e	Data source: <from 'dfdi<="" th=""><th></th><th></th></from>		
Itessages Itessage CompanyAddressChar (/workspaces/mm5/Mes										
nessage is a global element that models an entire document of data.										
	8							Name	Type	Value
Name		Type	Min Occurs	Max Occurs	Default Value	Sample Value		CompanyAddressChar		
🖃 e Comp	anyAddressChar							CompanyName	xs:string	My Co
= Se			1	1				Employee		
	CompanyName	string	1	1		a				
· · · ·	Employee		1	unbounded		-				
	sequence		1	1						
:	e EmpNo	intege		1		1				
	e Dept	intege		1		1				
	e Empname	string		1	body value3	body value3				
OFDL Processing Error Processing errors were encountered during parsing. You are advised to read the DFDL Trace to find out the root cause of this error. It may have been caused by previous processing errors, other than the final symptoms shown below. r Processing Errors CTDP3058E: Separator ',' not found at offset '102' for sequence within element '/CompanyAddressChar[1]/Employee[1]/Address[1]'. ParsedDataRegion[SimpleContent, startOffset = 87, length = 15, scd = #xscd[/schemaElement::CompanyAddressChar[1]/Employee:0/model::sequence/schemaElement::Employee/type::0/model::sequence										
To view the trac To view the par	e captured while run tial logical instance th	ning the DFDL pars at was created by	er, dick the Open DF	DL Trace View too k the Open DFDL I	-	<u>ere</u> . toolbar button, or click				

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.

🔁 Naviga or 🔝 H	Problems	arse 💫 🔄 DFDL Test - Ser	anze 🕞 DL Test - Trace			
DFDL Test - Parse	E: Runs the DFDL parser wi	th the provided physical input dat	a and selected message, and up	dates the logical instar	nce view with the result of the j	parse.
Status: \varTheta Parsin	g completed with processing	errors: Thu Apr 05 04:40:37 CD	<u>T 2012</u>			
_ Input						
Data: /MessageM	1odellingPrefixLib/Company.I	Prefix.Char.txt		▼ Browse	Message: CompanyAddress	Char (/MessageModellingPrefixLib/
2 Employ 3 Employe 4 Employe 5 Employe	ee (empNum=222222 ee (empNum=333333 ee (empNum=444444	11 dept= 500 empN 2 dept=500 empName 3 dept=310 empName 4 dept=230 empName	=James May Addr:1 =Richard Hammond A =Jeremy Clarkeson	523 The Cutt: Addr:1716 Gre Addr:22"Rose	ings07Chatham07CH eat Windmill06Lon e Cottage, Pea Dr	nam, Ont"07L3G 1H7 12 2PR tel=208-203- 1don06W2 3RJ tel=20 "10Gloucester08GL0 5London07NW1 1QT te:

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 🛛 🗖 🖻						
Test Parse Model Test Serialize Model Hide pro	-	sections Focus on se	ected Show quick outline Create logical instance	e		
1essages 📮 😰 🕆 🤴 🗱 🖬 — — — 🖾 Representation Properties (M= Variables) 🗄 Asserts and Discriminators						
message is a global element that models an entire	document of data		sequence		?	
Name	Туре	Min Occurs Max Occ	<type filter="" text=""></type>	📑 🛃 🗶 🙀 🗉 🛙	E 🖶	
e CompanyAddressChar			Property	Value	?	
		1 1	General			
E CompanyName	string	1 1	Data Format Reference	<default format=""></default>		
e Employee	-	1 unbound	Encoding (code page)	뤔 <dynamically set=""></dynamically>		
a sequence		1 1	Byte Order	🛃 <dynamically set=""></dynamically>		
EmpNo	integer	1 1	Ignore Case	뤔 no		
e Dept		1 1	Fill Byte	扂 0		
e Empname	-	1 1	Content			
e Address		1 1	Initiated Content	暑 no		
		1 1	Sequence Kind	뿸 ordered		
E StreetName		1 1	Occurrences			
e City	-	1 1	Min Occurs S	暑 1		
e ZipCode		1 1	Max Occurs S	晃 1		
	-		Alignment			
e Tel			Delimiters			
e Salary	decimal	1 1	Separator	昂,		
Add a Local Element			2 idator			

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	🛃 <no initiator=""></no>
	· · · ·

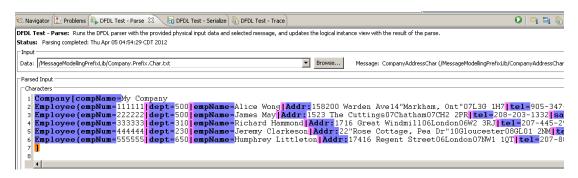
Save the model.

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

😤 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 <u>t</u>
1 Company [compName=My Company	
2 Employee (empNum=111111] dept=500 empName=Alid 3 Employee (empNum=222222 dept=500 empName=Jame	Do not display this message again
	hard Hammond Addr:1716 Great WindmillUbLondonUbW2 3RJ tel=207-4
	emy Clarkeson Addr: 22"Rose Cottage, Pea Dr"10Gloucester08GL01 2
	phrey Littleton Addr: 17416 Regent Street06London07NW1 1QT tel=2
7	
8	

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.

📑 DFDL Test - Logical Instance 🛛 🛛 🔓 🗌 🖼						
Data source: <from 'dfdl="" -="" parse'="" test="" view=""></from>						
Message: CompanyAddressChar (/workspaces/DFDL/MessageModellingPrefixLib/CompanyA						
Thessage. CompanyAddress	chai (monopace	sybi beyinessageinodelling	Frenzeby company a v			
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.

st Parse Mod	lel Test Serialize Model Hide proper	ties Show all se	ections Focus	on selected Sl	how quick outline Create logical instance		
■Messages	5 / 3 3 4 4 X	E . E		_	Representation Properties		
	a global element that models an entir		ata.		CompanyAddressBin (Element)		0
Name	V-	Туре	Min Occurs	Max Occurs	Þ	📑 🛃 🗙 🔆 🕀	
=	CompanyAddressBin				Property	Value	?
6	sequence		1	1	Comment S		
1	e CompanyName	string	1	1	General		
1	 Employee 		1	unbounded	Data Format Reference	<pre><default format=""></default></pre>	
	🖃 🚥 sequence		1	1	Encoding (code page)	₽ <dynamically set=""></dynamically>	
1	e EmpNo	integer	1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
1	e Dept	integer	1	1	Ignore Case	뤔 no	
1	e Empname	string	1	1	Fill Byte	昂 0	
1	E Address		1	1	Content		
	sequence		1	1	Length Kind	🖥 delimited	
	e StreetName	string	1	1	Occurrences		
	e City	string	1	1	Min Occurs S	<u> </u> 1	
	e ZipCode	string	1	1	Max Occurs S	晃 1	
1	e Tel	<string></string>	1	1	Alignment		
	e Salary	decimal	1	1	Delimiters		_
Add a Lo	cal Element		-	-	Initiator Terminator	Company[1%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.

- 8 🚺 CompanyAddressBin.xsd 🛛 Eô E A Έį. E Test Parse Model Test Serialize Model Hide properties Show all sections Focus on selected Show quick outline Create logical instance Representation Properties 🔍 🕪 = Variables 📄 🔚 Asserts and Discriminators 💂 🛛 û û 🗸 🗶 🖡 🔓 ▼Messages ? A message is a global element that models an entire document of data sequence 📑 🔜 🗙 🔆 🖻 🖻 🖶 Min Occurs Max Occurs Name Type 🖃 🖻 CompanyAddressBin Property ? Value 🖃 🚥 sequence General e CompanyName Data Format Reference <default format> string Ra <dynamically set> Ra <dynamically set> Encoding (code page) 🖃 🖻 Employee unbounded Byte Order 🖃 🚥 sequence 1 , no e EmpNo integer Ignore Case 昂0 e Dept Fill Byte integer 1 Content e Empname string 1 Initiated Content 뤎 no 🖃 🖻 Address Sequence Kind 🖥 ordered e sequence Occurrences e StreetName string 1 累 1 累 1 Min Occurs S e City string Max Occurs S e ZipCode string Alignment e Tel <string> 1 1 Delimiters e Salary decimal 1 1 + Separator **R**, Add a Local Element

 R
 <no initiator >

 R
 <no terminato</td>

 Initiator Terminator <no terminator> Output New Line 暑 %CR;%LF;

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

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	E Parse Model	E) Test Serialize Model	Hide properties	A Show all sec	tions Focus
I —	lessages nessage is a	हा हा थि global element that m	1	ocument of dat	ta.
	Name			Туре	Min Occurs
	- e (CompanyAddressBin			
		sequence			1
	1	e CompanyName		string	1
		🖃 🖻 Employee			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
Name:	TwoBytesBin
Inherit from:	short
	OK Cancel

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

Parse Model Test Serialize Model Hide proper	ties Hide empt	y sections Fo	cus
Name	Туре	Min Occurs	Ma
🖃 🖻 CompanyAddressBin			
sequence		1	1
e CompanyName	string	1	1
E 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 *C	🖸 *CompanyAddressBin.xsd 🖾 🗁 🗗								
Test F	Parse Model Test Serialize M	Iodel Hide propertie	s Hide empt	sections	Focus on select	ed	Show quick outline Create logical instar	nce	
	e Em	pname	string	1	1		Representation Properties (X)=	Variables	
	E e Ad	dress		1	1				0
		sequence		1	1		TwoBytesBin (Type)		0
		e StreetName	string	1	1				
		e City	string	1	1		<type filter="" text=""></type>	李 昂	🗱 🔆 🙀 🖽 🖽 🖶
		e ZipCode	string	1	1		Property	Value	 ?
	e Tel		<string></string>	1	1		Comment S		
	e Sal	ary	decimal	1	1		General		
	Add a Local Element						Data Format Reference	<pre><default format=""></default></pre>	
							Encoding (code page)	🛃 <dynamically set=""></dynamically>	
ŀG	lobal Elements (0 eleme	nts)					Byte Order	🛃 <dynamically set=""></dynamically>	
Ad	lobal element represents a na	amed instance of a co	omplex or simp	le type.			Ignore Case	昂 no	
							Fill Byte	晃 0	
+0	omplex Types (0 comple	ex types)					Content	short	
An	omplex type defines the elem	ents and groups that	t represent a	structure			Representation	뿸 text	
	omplex type dennes are elen	iento una groupo una	erepresent at	ra accar cr			Length Kind	🔁 delimited	
-5	imple Types 🛛 戻	x				1	Text Content		
			ara simple ale	monto			 Number Representation 	🔁 standard	
AS	A simple type defines the allowed values for one or more simple elements.				Number Justification	🛃 right			
	Name	Base Type					Number Pad Character	뤔 %SP;	
l r							Pad Kind	🛃 none	
1 1	TwoBytesBin	short					Trim Kind	晃 padChar	
							Escano Schomo Doference	🔲 rocConEioldeEmtiDoce	edEacanaSchar

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)

	e Address		1	1	Content string	
	🖃 🚥 sequence		1	1	Representation 🔤 text	
:	e StreetName	string	1	1	Length Kind prefixed	
:	e City	string	1	1	Length Units bytes	
	e ZipCode	string	1	1	Prefix Length Type TwoBytesBin	
:	e Tel	<string></string>	1	1	Prefix Includes Prefix Lengt yes	
	e Salary	decimal	1	1	Nillable 😒 🔤 🛃 false	
					Default Value 🖻 🦷 currents	

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).

Test Parse Model Test Serialize Model Hide prop	erties Hide em	A pty sections	Focus on selecte	d !	5how quick outline	Create logical	instance			
▼Messages 🔄 😣 û 🤴 💥	Messages 🖉 🖟 🕀 💥 🖺 🖬							es 🖹 Asserts and Di	scriminators	
A message is a global element that models an entire document of data.					sequence					?
Name	Туре	Min Occurs	Max Occurs		<type filter="" text=""></type>			参 見	× %	⊞ ⊟ ♣
🖃 🖻 CompanyAddressBin					Property			alue	**	(?)
e seguence		1	1		General		v,	auc		
E CompanyName	string	1	1			nat Reference	<	lefault format>		
: e Employee		1	unbounded		Encoding	(code page)	見	<dynamically set=""></dynamically>		
e e sequence		1	1		Byte Orde	er		<dynamically set=""></dynamically>		
EmpNo	integer	1	1		Ignore Ca	ase		no		
e Dept	integer	1	1		Fill Byte		묽	0		
e Empname	string	1	1		Content					
E e Address		1	1		Initiated	Content		no		
		1	1		Sequence		묽	ordered		
: e StreetName	string	1	1		Occurrence					
: e City	string	1	1		Min Occur		2			
: e ZipCode	string	1	1		Max Occu	urs S	묽	1		
E Tel	<string></string>	1	1		Alignment					
e Salary	decimal	1	1		Delimiters		-			
Add a Local Element					⊕ Separato	r	2	1		
					Torminato		57			
					Torminate	nr.		con terminator's		

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>
	F=1

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 Develo	🌐 Test Parse I	Model	
File	Edit Navigate Se			-
Ē2	- 11 🖪 🖆 😭	Select mes	File Selection	
е [] <mark>2</mark> е 22	CompanyAddress Test Parse Model 1 •Messages A message is a globe	Parser Inc Select con C Contei C Contei Input file r	MessageModellingPrefixLib Company.Prefix.Bin.PrefixIndudingLength.txt Company.Prefix.Bin.txt Company.Prefix.Char.txt ibirary.descriptor	Browse
88 88	Name e Comp 	Specify runt Runtime e Provide ru Encoding (Floating p		be dynamically set. More
			Select an input file from the file system Browse,	
		Restore Defa	OK Cancel	

17. Close the yellow completion pop-up.

🗞- Navigator 🖹 Problems 🕒 DFDL Test - Parse 🛛 📘 DFDL Test - Serialize 🗈 DFDL	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=Alio	Do not display this message again
3 Employee (empNum=222222 dept=500 empName=Jame	
	ard Hammond Addr:1716 Great WindmillU6LondonU6W2 3RJ tel=207-4
	emy Clarkeson Addr:22"Rose Cottage, Pea Dr"10Gloucester08 <u>GL01</u> 2
6 Employee (empNum=555555 dept=650 empName=Hum)	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.txt 💌 Browse Message: CompanyAddressBin (MessageModellingPrefixLib/CompanyAddressBin (Mess	mpanyAddressBin.xsd)
Parsed Input	
Characters	
1 Company [compName=My Company	
2 Employee (empNum=111111 dept=500 empName=Alice Wong Addr 10200 Warden Avell Warkham, Ont 10136 1H7 tea	
3 Employee (empNum=222222 dept=500 empName=James May Addr: 023 The CuttingsIIChathamIICH2 2PR tel=208-2 4 Employee (empNum=333333 dept=310 empName=Richard Hammond Addr: 016 Great WindmillIILondon IW2 3RJ tel	
* Employee (employme = 535353 dept= 510 employme = Archard Hardman and and the Great white in the fore the state of the	207-445-2955 Sdl= 5
6 Gloucester [GL01 2XN] tel 743-123-4567 sal 75599.95]	
7 Employee (empNum=555555 dept=650 empName=Humphrey Littleton Addr: W411 Regent Street London NW1 10T	tel=207-883-1238 s
9	

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 🛛	
Data source: <from 'dfdl'<="" th=""><th>Test - Parse' view</th><th>1></th></from>	Test - Parse' view	1>
Accesses CompanyAddress		/mmE/MassaseMadallineDraft
ressage: CompanyAddress	bin (/workspaces	/mm5/MessageModellingPrefix
ree 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