IBM Integration Bus

Message Modeling with DFDL

Lab 5 Using DFDL length prefixes

June, 2013

Hands-on lab built at product code level Version 9.0.0.0

1.	INT	RODUCTION	3
	1.1 1.2	LAB PREPARATION LAB SCENARIO	3 3
2.	IMP	ORT THE BASE MODEL	4
3.	CRE	EATE THE PREFIX LENGTH CHARACTER SCENARIO	8
4.	CRE	EATE THE PREFIX LENGTH BINARY SCENARIO	22

1. Introduction

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

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:\student directory. This will create a subdirectory called MessageModelling, with several further subdirectories. If you are using the pre-supplied vmware image, this will already be available.

1.2 Lab Scenario

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:\student\MessageModelling\prefixes\PrefixLabStartingPoint.zip.



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



🜔 Rename Resource		×
Enter the new resource name:		
CompanyAddressCharlxsd		
	OK Can	ncel



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	4	E	€Ę}	\bigtriangledown	
Application Development					New
MessageModellingPrefixLib Schema Definitions Gradie (default namespace) CompanyAddressBin.xsd CompanyAddressChar.xsd Hrdie http://www.ibm.com/dfdl/RecordSeparatedi Other Resources	FieldFormat				

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

▼Messages Image: I	
Name Type Min Occurs Max	
	< Occurs
🖃 🖻 CompanyAddressBin	
🖃 🚥 sequence 1 1	
E CompanyName string 1 1	
: 主 e Employee 1 unb	ounded
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.

★Messa A messa	ges 🛛 🙀 🛊 국 je is a global element that model	אן 🗱 🛛 🔄 Is an entire do	cument of da	ata.	
Nam	э	Туре	Min Occurs	Max Occurs	
	e CompanyAddressChar				
	🖃 🚥 sequence		1	1	
	e CompanyName	string	1	1	
	표 🖻 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

1. Open and expand the CompanyAddressChar.xsd message model.

Parse Model	Test Serialize Model Hide prop	erties Show ba	aic Show all se	ections	Focus on selected Show quick outline	Create logical instance	
Messages					Representation Properties	= Variables (not supported in current IB	BM I
message is a (global element that models an en	ire document of a	data.		CompanyAddressChar (Elemen	t)	
Name		Туре	Min Occurs	Max	<type filter="" text=""></type>	× 🔌 昆	Ī
🗆 e 🛙	IompanyAddressChar				Property	Value	
	••• sequence		1	1	Comment S		
1	e CompanyName	string	1	1	🖃 General		
12	🖃 e Employee		1	unbo	Data Format Reference	<pre><default format=""></default></pre>	
	🖃 🚥 sequence		1	1	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
12	e EmpNo	integer	1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
	e Dept	integer	1	1	Ignore Case	🛃 no	
12	e Empname	string	1	1	Fill Byte	昂。	
1	E e Address		1	1	Content		
			1	1	Length Kind	ᅯ delimited	
1	e StreetName	string	1	1	Occurrences		
-	e City	string	1	1	Min Occurs S	<u> </u>	_
		string	1	1	Max Occurs S	<u>昂</u> 1	
1		string	1	4	Alignment		
		<scring></scring>		1	Delimiters	-	
i i i i i i i i i i i i i i i i i i i	e balary	decimal	1	1	Initiator	Company[
Add a Local	<u>i Liemenc</u>				Terminator]%CR;%LF;	÷
					Empty Value Delimiter Policy	/ 🛃 initiator	_
					Output New Line	1 %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.

arse Model Test Serialize Model Hide prope	rties Show ba	aic Show all se	ctions F	ocus on selected Show quick outline	e Create logical instance	
essages 🛛 🙀 🛊 🛠 🙀	E E		-	Representation Proper	Asserts and Discrimin 🗱= Variables (r	not si
essage is a global element that models an entir	e document of (data.		sequence		
Name	Туре	Min Occurs	Max	<type filter="" text=""></type>	* %]]]
🖃 🖲 CompanyAddressChar				Property	Value	
🖃 🚥 sequence		1	1	🖃 General		
e CompanyName	string	1	1	Data Format Reference	<pre><default format=""></default></pre>	
e Employee		1	unbo	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
		1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
e EmpNo	integer	1	1	Ignore Case	뤎 no	
e Dept	integer	1	1	Fill Byte	昂이	
e Empname	string	1	1	Content		
		1	1	Initiated Content	昂 no	
		1	1	Sequence Kind	🛃 ordered	
: StreetName	chripa	1	1	Alignment		
	string	-	1	Delimiters		
	suning	1	1		昂,	
e zipCode	string	1	1	Initiator	॑॑॑॑॑॑॑॑	
e Tel	<string></string>	1	1		🛃 <no terminator=""></no>	
e Salary	decimal	1	1	Output New Line	🛃 %CR;%LF;	

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.

0 Co	ompanyAddressChar.xsd 🛛						
Test	Parse Model Test Serialize Model Hide properti	as Show basic	A Show all se	ctions	Foc	us on selected Show quick outline	Create logical instance
-	1essages 🛛 🗐 🕆 🖑 🕱 🔰	B B				🗌 Representation Proper 🛛 📘 A:	sserts and Discrimin 🔀=
Ar	nessage is a global element that models an entire of	document of dal	ta.			equence	
	Name	Туре	Min Occurs	Max		<type filter="" text=""></type>	3
	🖃 🖻 CompanyAddressChar					Property	Value
	🖃 🚥 sequence		1	1		🖃 General	
	e CompanyName	string	1	1		Data Format Reference	<pre><default format=""></default></pre>
	E Employee		1	unbo		Encoding (code page)	🔒 <dynamically set=""></dynamically>
	sequence		1	1		Byte Order	🛛 🛃 <dynamically set=""></dynamically>
	e EmpNo	integer	1	1		Ignore Case	昂 no
	e Dept	integer	1	1		Fill Byte	昂0
	Emphame	strina	1	1		Content	
			1	à l		Initiated Content	昂 no
			1	1		Sequence Kind	🛃 ordered
		abuta a				🕀 Alignment	
		string	1	1		Delimiters	
	e City	string	1	1			累,
	e ZipCode	string	1	1		Initiator	🛃 <no initiator=""></no>
	e Tel	<string></string>	1	1		🛨 Terminator	🛃 <no terminator=""></no>
	e Salary	decimal	1	1		Output New Line	🛃 %CR;%LF;
	Add a Local Element						

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.



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	ole Type		<u>- 🗆 ×</u>
Name:	TwoCharsText		
Inherit from:	short		▼
		ок	Cancel

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

CompanyAdo	dressBin.xsd 🚺 *CompanyAddr	essChar.xsd	X			
E Test Parse Mod	el Test Serialize Model Hide proper	ties Show ad	≎ vanced Hide e	A mpty section	Focus on selected Show quick outline Create logical in:	stance
▼Schema			Ba ta El		Representation Properties (*)= Variables (not supp	orted in current IBM DFDL implementation)
· Schema	∲n ¢⊓ ¢n ¢n ¢n ¢n ¢n é	20 49 4 -	8- 11- 115		Two(barsText (Type)	· · · · · · · · · · · · · · · · · · ·
Namespac	e <null namespace=""></null>					
▶Schema	a References (0 includes, 1 impo	rt)			<type filter="" text=""></type>	
A schema	file in the same namespace uses an inc	lude. A schem	a file in a differe	ent namesp	Property	Value
					Comment	
▼Messages	· · · · · · · · · · · · · · · · · · ·	Es En			🖃 General	
A messarie is	a global element that models an entire	document of	data.		Encoding (code page)	🛃 <dynamically set=""></dynamically>
H mossage is	a global olement chat models an entire	accument of	aata.		Byte Order	🛃 <dynamically set=""></dynamically>
Name		Type	Min Occurs	Max Occi	Content	short
		1700	Thin Occurs	Fidx occ	Representation	🛃 text
	CompanyAddressChar				Length Kind	₽ delimited
		abola a	1	1	Text Content	
		string	1	1 uniternal	Text Number Representation	Standard
				unbound	Escape Scheme Reference	Fa recoeptieldshmt:RecordEs
		internet.	1	1	- Deliniters	
1	E Empirio	integer	1	1	Terminator	
	E Dept	integer	1	1	T Validation	short
		string	1	1	- Validación	Shore
:			1	1		
	e e sequence	1.5	1	1		
	e Streetvane	string		1		
		string	1	1		
		string	1	1		
1		<suning></suning>	1	1		
: Add a Lo	E Salary	decimal	1	1		
Muu a Lu	<u>car ciemenc</u>					
⊁Global Ele	ments (0 elements)					
A slabal slave	menes (o ciemenes)		ala buas			
A giobal elem	ient represents a named instance or a	complex or sim	pie cype.			
€Compley *	Types (0 complex types)					
A complex by	ne defines the elements and around the	at represent -	chruchure		1	
A COMPLEX BY	pe dennes une elements and groups (n	iac represent a	isciduture.			
▼Simple Tv	rpes 🖉 🕱					
A simple type	e defines the allowed values for one or	more simple el	ements			
in sample cype		more simple of	on on the second			
Name	Base Type					
	TwoCharsText short					
				_	llar	

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.

0 *CompanyAddr	essChar.xsd 🛛								
E	Eŭ 💷		A		Ë	Ē			Ī
Test Parse Model	Test Serialize Model Hide prop	erties Show ba:	sic Show all sect	tions Focus on selected	Show quick outline	Create logical instance			
	e ZipCode	string	1	🔲 Representation Prop	erties 🛛 🗱 Variat	les (not supported in curren	t IBM DFDL implement	tation)	
	e Tel	<string></string>	1	TwoBytesChar (Type)				
1	e Salary	decimal	1						-
Add a Local El	ement			<type filter="" text=""></type>			🔜 🗙 🔆 📗	昂 6	Ŧ
				Property		Value		?	T
▼Simple Types	📮 🗶			Comment 🛐					Τ
A simple type defin	nes the allowed values for one or	more simple elem	ients.	🖃 General					
				Data Format R	eference	<pre><default format=""></default></pre>			
Name	Base Type			Encoding (code	e page)	🛃 <dynamically set=""></dynamically>			
= Two	BytesChar integer			Byte Order		🛃 <dynamically set=""></dynamically>			
				Ignore Case		🔁 no			
▶Data Formats	(1 format)			Fill Byte		昇 0			
A data format is a	container of DFDL properties.			Content		integer			
				Representation	ו	🛃 text			
→Variables (4 v	ariables)			Length Kind		暑 delimited			
A variable holds a	value that can be used in DFDL e	xpressions.		Text Content					
				∓ Text Number R	epresentation	🚬 standard			
				Number Justific	ation	🛃 right			

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	integer
Representation	暑 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.

- Content short 🛃 text Representation explicit 🖃 Length Kind Length 2 Length Units 晃 characters 😑 Text Content Text Number Representation 67 Number Pattern <unset> Grouping Separator 67, 昂. Decimal Separator Escape Scheme Reference 🛃 recSepFieldsFmt:RecordEscap Delimiters
- 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	晃 text
🖃 Length Kind	explicit
Length	2
Length Units	🛃 characters
E Text Content	
Text Number Representation	🔁 standard
Number Pattern	00
Grouping Separator	品,
Decimal Separator	
Escape Scheme Reference	🛃 recSepFieldsFmt:RecordEscap

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.

🛈 CompanyAddressChar.xsd 🛛						
Test Parse Model Test Serialize Model Hide prop	erties Show basic Show	Letter Create logical instance				
•Messages 🛛 🐺 🕆 🦊 💥	E B	🔺 🔲 Representation Properties 🛛 📄 Asserts and Discriminators 🗱 🗤	/ariables			
A message is a global element that models an entire	document of data.	StreetName (Element)				
Name	Type Min Occu	<pre>rs </pre> <type filter="" text=""></type>				
🖃 🖻 CompanyAddressChar		Property Value				
sequence	1	Comment S				
e CompanyName	string 1	🕞 General				
: e Employee	1	Data Format Reference 				
🖃 🚥 sequence	1	Encoding (code page)				
: EmpNo	integer 1	Byte Order 🛛 🛃 <dynamically set=""></dynamically>				
: Dept	integer 1	Ignore Case 🛃 no				
Empname	string 1	Fill Byte 🛃 0				
😑 😑 Address	1	Content string				
🖃 🚥 sequence	1	Representation 🛃 text				
e StreetName	string 1	Length Kind 🛃 delimited				
e City	string 1	→ Nillable S Raise				
e ZipCode	string 1	Default Value S <unset></unset>				
e Tel	<string> 1</string>	Fixed Value S <unset></unset>				
E Salary	decimal 1	Text Content				
		String Justification 🖳 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.

Property	Value	?
Comment 🛐		
🖃 General		
Data Format Reference	<pre><default format=""></default></pre>	
Encoding (code page)	🛃 <dynamically set=""></dynamically>	
Byte Order	🛃 <dynamically set=""></dynamically>	
Ignore Case	뤎 no	
Fill Byte	昇0	
Content	string	
Representation	🛃 text	
🖃 Length Kind	prefixed	
Length Units	🛃 characters	
Prefix Length Type	TwoCharsText	
Prefix Includes Prefix Length	<u>no</u> 🔽	
Nillable S	🛃 false	
Default Value 📓	<unset></unset>	
Fixed Value 💲	<unset></unset>	

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

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.

Test Parse Indel Test Serialize Model Hide properties Show basic Show all sections Focus on selected Show quick outline Creat Namespace (-null namespace) Change namespace Schema file namespace uses an indude. A schema file in a different namespace uses an import. Test Parse Induct Test Selection State (State (Sta	CompanyA	ddressChar.xsd 🗙	
Test Partie inder Test Partie inder Schwaszer inder inde properties Snow dask show all sectors Change namespace Change namespace Namespace Change namespace Change namespace ZipCode A schema file in the same namespace uses an include. A schema file in a different namespace uses an include. The same namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different namespace uses an include. A schema file in a different	E.		
Schema References Image: Schema Referenc	Namespa	ce <null namespace=""></null>	Change namespar
A schema file the same namespace uses an include. A schema file in a different namespace uses an import. Text Parse Model Text Parse Model Text Parse Model Text CompanyAddressChar Parser Input Select content to be parsed against schema. Content from DFDL Test - Serialize' view Content from a data file Trput file name: Citytudent(MessageModeling)prefixes)Company.Prefix.Char.txt File Selection Select an input file Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Compan	▼Schen	na References 🛛 🕱 🕱	ZinCada
Verified and a conder sounds difficitier sounds and if the conder sounds and if the conder sounds difficitier sounds and the conder sound and the conder	A schem	a file n the same namespace uses an include. A schema file in a diffe	erent namespace uses an import.
Message Select message for testing. More Message name:* CompanyAddressChar Parser Input Select content to be parsed against schema. Content from DFDL Test - Serialize' view Content from DFDL Test - Serialize' view Content from a data file Input file name:* Citystudent/MessageModeling/prefixes/Company.Prefix.Char.bxt Browse Select an input file: MessageModelingPrefix.bit MessageModelingPrefix.bit MessageModelingPrefix.bit MessageModelingPrefix.Char.bit MessageModelingPrefix.bit MessageModelingPrefix.Char.bit Company.Prefix.Char.bit MessageModelingPrefix.char.bit Company.Prefix.Char.bit Company.Prefix.Char.bit Company.Prefix.Char.bit Select an input file from the file system Descrit Select an input file from the file system	Т	est Parse Model	
A m Select message for testing. <u>More</u> Message name:* CompanyAddressChar Parser Input Select content to be parsed against schema. Content from DPD. Test - Serialize' view Content from a data file Input file name:* C:\student\MessageModeling\prefixeS\Company.Prefix.Char.txt File Selection Select an input file: Company.Prefix.Char.txt More Select an input file: Company.Prefix.Char.txt More Select an input file from the file system Provse MessageModelingPrefix.Char.txt More More Company.Prefix.Char.txt More Company.Prefix.Char.txt	T M		
Message name: * CompanyAddressChar Parser Input Select content to be parsed against schema. Content from 'DFDL Test - Serialize' view Content from a data file Input file name: * Clystudent\MessageModeling\prefixes\Company.Prefix.Char.txt File Selection Select an input file: Company.Prefix.Bin.txt Company.Prefix.Bin.txt Company.Prefix.Char.txt Company.Prefix.Char.txt MessageModeling\Prefix.Bin.txt Company.Prefix.Char.txt Select an input file from the file system Provse OK	Am Se	essage elect message for testing. <u>More</u>	
Parser Input Select content to be parsed against schema. Content from 'DFDL Test - Serialize' view Content from a data file Input file name:* C:\student\MessageModeling\prefixes\Company.Prefix.Char.txt Tele Selection Select an input file: MessageModelingPrefix.bin.txt Company.Prefix.Char.txt TBMdefined Company.Prefix.Char.txt Select an input file from the file system Browse OK Cancel	M	essage name:* CompanyAddressChar	•
Select content to be parsed against schema. C Content from 'DFDL Test - Serialize' view C Content from a data file Input file name:* C:\student\MessageModeling\prefixes\Company.Prefix.Char.txt Prowse Select an input file Company.Prefix.Bin.txt Company.Prefix.Char.txt C Content file from the file system C Company.Prefix.Char.txt C Cancel C C Content file from the file system C C Content file from the file system C C C C C C C C C C C C C C C C C C C	⊢ Pa	arser Input	
Content from DPDL Test - Serialize' view Content from a data file Input file name:* C:(student/MessageModeling\prefixes\Company.Prefix.Char.txt	Se	elect content to be parsed against schema.	
Select an input file from the file system Select an input file from the file system Select an input file from the file system OK Cancel		Content from DFDL Test - Serialize' view	
File Selection Select an input file: MessageModellingPrefix.lb BMdefined Company.Prefix.Bin.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Select an input file from the file system Descrit Select an input file from the file system	In	put file name:* C:\student\MessageModeling\prefixes\Company.F	Prefix.Char.txt
Select an input file: Select an input file: MessageModellingPrefix.lb Company.Prefix.Char.txt Company.Prefix.Char.txt Company.Prefix.Char.txt Select an input file from the file system Ditems Select an input file from the file system Drowse OK	6	File Selection	
Select an input file from the file system Browse OK OK Cancel	5	Select an input file:	
Si R Image: Signature of the system Image: Select an input file from the file system Browse OK	ſ	🗆 🧀 MessageModellingPrefixLib	
Select an input file from the file system Browsen OK Cancel		E BMdefined	e model to be dynamically set. More
Si R Pro Ditems Descri		Company.Prefix.Char.txt	
✓Si R Image: Product of the system Browse Description Browse			
Si R Pro Ditems Description Select an input file from the file system Browse, OK Cancel			
Si R Pro Ditems Descrit Concel	ſ		
Select an input file from the file system Descri OK			
Pro Ditems Descri OK Cancel OK Cancel			
Pro Ditems Descri Descri OK Cancel			
Ditems Descrip Descrip Descrip OK Cancel	Pro		
Descri	0 items	Select an input file from the file system	
OK Cancel	Descrij	Browse	
OK Cancel			
			OK Cancel
		(?) OK Cancel	

16. Success !

(Or perhaps not)

CompanyAddressBin.xsd CompanyAddressChar
Test Parse Model Test Serialize Model Show properties Show basic Show all sections Focus on selected Show quick outline Create logical instance Data source: <from 'dfdl="" 1<="" td=""> Vesterma Image: Image:</from>
Schema Schem
Namespace Chanae namespace > Schema References (0 includes, 1 import) A schema file in the same namespace uses an include. A schema file in a different namespace uses an import. > Messages > • • • • • • • • • • • • • • • • • •
Image (with namespace) Image namespace (with namespace) • Schema References (0 includes, 1 import) A schema File in the same namespace uses an include. A schema file in a different namespace uses an import. • CompanyAddressChar CompanyAddressChar CompanyAddressChar • Messages • • • • • • • • • • • • • • •
>Schema References (0 includes, 1 import) A schema file in the same namespace uses an include. A schema file in a different namespace uses an import. Mame Type Min Occurs Max Occurs Default Value Sample Value
A schema file in the same namespace uses an include. A schema file in a different namespace uses an import.
Image:
Image: Sequence 1 Image
A message is a global element that models an entire document of data. Name Type Min Occurs Max Occurs Default Value Sample Value
Name Type Min Occurs Max Occurs Default Value Sample Value
Name Type Min Occurs Max Occurs Default Value Sample Value CompanyAddressChar Sequence 1 CompanyName String I I Employee I I
 © CompanyAddressChar © expanyAddressChar © CompanyAame 1 1 a
Image: Company Name 1 1 Image: Company Name string 1 1 Image: Company Name 1 1 1 Image: Company Name string 1 1
i CompanyName string 1 a i i Employee 1 unbounded i i Employee 1 1 i E Employee 1 1 1
Image: Constraint of the set of
Image: sequence 1 1
i @ EmpNo integer 1 1 i @ Dept integer 1 1 i @ Dept integer 1 1
Image: Comparison of the second sec
Emphane string 1 1 hody value?
© DEDL Processing Error
Processing errors were encountered during parsing.
Too are autised to read the OPDL mate to find out the root cause of this error. It may have been caused by previous processing errors, outer than the final symptoms shown below.
CTDP3058E: Separator ',' not found at offset '102' for sequence or choice within element '/CompanyAddressChar[1]/Employee[1]/Address[1]'.
ParsedDacakegion[SimpleConcent, starcOrrset = 87, length = 15, scd = #xscd(schematlement::CompanyAddressChar/type::U/model::sequence/schematlement::tmployee/type::U/m
 Errors received during parsing are highlighted in the parsed input section of the DFDL Test - Parse view and hover help is provided.
 To view the trace captured while running the DFDL parser, click the Open DFDL Trace View toolbar button, or click <u>here</u>.
 To view the partial logical instance that was created by the DFDL parser, click the Open DFDL Logical Instance View toolbar button, or click here.
The view menu on the view toolbar provides options to control how the data is displayed in the view. Click the arrow icon on the toolbar or here to open the menu.
Do not 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.

Sa. Naviga or 🖾 Problems 🕞 DFDL Test - Parse 🐼 🕞 DFDL Test - Serialize 🕞 D						
DFDL Tes - Parse: Runs the DFDL parser with the provided physical input data and select d message, and updates the logical instance view with the result of the parse.						
Status: IP Parsing completed with processing errors: Thu Apr 05 04:40:37 CDT 2012						
Input Data: MessageModellingPrefixLib/Company.Prefix.Char.txt Browse Message: CompanyAddressChar (/MessageModellingPrefixLib/						
Parsed Input						
Characters						
1 Company compName=My Company						
2 Employee (employee and a second sec						
4 Employee (empNum=333333] dept=310 empName=Richard Hammond Addr: 1716 Great Windmill06London06W2 3RJ tel=20						
5 Employee(empNum=444444 dept=230 empName=Jeremy Clarkeson Addr:22"Rose Cottage, Pea Dr"10Gloucester08GL0						
6 Employee (empNum=555555) dept=650 empName=Humphrey Littleton Addr:17416 Regent Street06London07NW1 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.

CompanyAddressCha	ar.xsd 🛛						
E Parse Model Tes	t Serialize Model Hide proper	ties Show bas	ic Show all se	ctions Focus on s	l elected Sh	ti now quick outline Create log	E jical instance
Messages	동 문 순 · · · · · · · · · · · · · · · · · ·		lata.		seque	presentation Proper	Asserts and Discrimin
message is a global		accament or e			<type< td=""><td>e filter text></td><td></td></type<>	e filter text>	
Name		Туре	Min Occurs	Max Occurs	Prope	erty	Value
🖃 🖻 Compa	anyAddressChar				🗆 G	eneral	
🖃 🚥 sec	quence		1	1		Data Format Reference	<pre><default format=""></default></pre>
: e	CompanyName	string	1	1		Encoding (code page)	🛛 🛃 <dynamically set:<="" td=""></dynamically>
	Employee		1	unbounded		Byte Order	🛃 <dynamically set:<="" td=""></dynamically>
			1	1		Ignore Case	🛃 no
	Emphis	integer	1	1		Fill Byte	累。
		inceger	1	-	🗆 🗆 C	ontent	
	e Dept	inceger	1	1		Initiated Content	🛃 no
1	e Empname	string	1	1		Sequence Kind	🛃 ordered
	e Address		1	1	± A	lignment	
	🖃 🚥 <mark>sequence</mark>		1	1		alizzitara	
0	e StreetName	string	1	1	E	Separator	累,
1	e City	string	1	1		T-iki-k	the initiatory
1	e ZipCode	string	1	1	E	E Terminator	💂 <no terminator=""></no>
1	e Tel	<string></string>	1	1		Output New Line	💂 %CR;%LF;
1	e Salary	decimal	1	1			
Add a Local Eleme	ent						

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=Alic	Do not display this message again
3 Employee(empNum=2222221dept=500)empName=Jame	and Hammond 1 Addm 1 // 16 (most Windmi 1 116) and an 116W? 711 1 to 1 = 7117 /
5 Employee (empNum=444444 dept=230 empName=lerg	mara Hammond Addr. 1716 Great WindmillooLondonoow2 SKJ ter-207-4
6 Employee (empNum=5555551 dept=6501 empNume=Humr	phrey Littleton Addr: 17416 Regent Street06London07NW1 10T tel= 2
8	

20. Close the yellow completion pop-up.

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

😪 Navigator 🔝 Problems 🛱 DFDL Test - Parse 🛛 🕞 DFDL Test - Serialize) 🚯 DFDL Test - Trace	🗉 🕞
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 04:54:29 CDT 2012	
_ Input	
Data: [MessageModellingPrefix.b]/Company.Prefix.Char.bxt 💌 Browse Message: CompanyAddressChar ([MessageModellingPrefix.b]/CompanyA	ddressChar
Parsed Input	
Characters	
1 Company [compName=My Company	
2 Employee (empNum=111111 dept=500 empName=Alice Wong Addr: 158200 Warden Ave14 "Markham, Ont"07L3G 1H7 tel=90	35-347-
3 Employee (empNum=222222 dept=500 empName=James May Addr: 1523 The Cuttings07Chatham07CH2 2PR tel=208-203-1	332 <mark> sa</mark>
4 Employee (empNum=333333] dept=310 empName=Richard Hammond Addr:1716 Great Windmill06London06W2 3RJ tel=207	-445-2'
5 Employee (empNum=444444 dept=230 empName=Jeremy Clarkeson Addr: 22"Rose Cottage, Pea Dr"10Gloucester08GL01	2NM <mark> te</mark>
6 Employee (empNum=555555 dept=650 empName=Humphrey Littleton Addr:17416 Regent Street06London07NW1 1QT tel	207-8
8	

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 Instanc	e X		🖥 🗟 🗖 🗖
Data source: <from 'dfdl'<="" td=""><td>Test - Parse' view:</td><td>></td><td></td></from>	Test - Parse' view:	>	
Message: CompanyAddress	Char (/workspace:	s/DFDL/MessageModelling	PrefixLib/CompanyAı
Tree View XML View			
Name	Туре	Value	<u>▲</u>
🖃 CompanyAddressChar			
CompanyName	xs:string	My Company	
🖃 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.

E Parse Model Test S	6 5erialize Model	III Hide propertie	⇒ s Show basi	A Show all se	ctions Focus	ion selected S	ta how quick outlir	ie Create logical instance	
Messages	副 🚑 介 lement that mo	🖑 💥 [dels an entire d	locument of da	ata.				Representation Propertion CompanyAddressBin (Electronic)	es (x)= Variables (no ement)
Name			Туре	Min Occurs	Max Occurs			<type filter="" text=""></type>	× %
🖃 🖻 Compan	yAddressBin							Property	Value
🖃 🚥 segu	ence			1	1			Comment 🛐	
: e C	ompanyName		string	1	1			🖃 General	
: 🗆 e E	mployee			1	unbounded			Data Format Refer	<default format=""></default>
	sequence			1	1			Encoding (code pag	🛃 <dynamically set=""></dynamically>
1	e EmpNo		integer	1	1			Byte Order	🛃 <dynamically set=""></dynamically>
1	e Dept		integer	1	1			Ignore Case	昇 no
1	e Empname	•	string	1	1			Fill Byte	累 0
1	E Address		-	1	1			Content	
	= seque	nce		1	1			Length Kind	嚣 delimited
1	e St	reetName	string	1	1			Occurrences	—
	e Ci	ty	string	1	1			Min Occurs S	高·
1	e Zi	Code	strina	1	1			Max Occurs S	76 ¹
1	e Tel		<string></string>	1	1			Aignment Delimiters	
	e Salarv		decimal	1	1			- Demnicers	Company
				-			I	L Indoor	CombanAF

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

🚺 CompanyAddressBin.xsd 🛛	3						
Test Parse Model Test Serial	ize Model Hide properti	⇒ es Show basic	A Show all se	ctions Focus on se	E elected Show quick out	ine Create logical instance	
•Messages 🛛 🛃	🛃 🕆 🖑 💥 📔	tocument of da	+ 2			Representation Prope	E Asserts and Discrim
H message is a global ciciliei	ic chac models an entre i	accument of da	ca.			sequence	
Name		Туре	Min Occurs	Max Occurs		<type filter="" text=""></type>	📉 🗙 🔆 🗜
🖃 🖻 CompanyAdd	ressBin					Property	Value
🖃 🚥 sequence			1	1		🖃 General	
e Compa	anyName	string	1	1		Data Format Refer	<pre><default format=""></default></pre>
Employ	yee		1	unbounded		Encoding (code pa	Realized the set of th
🖃 🚥 Se	quence		1	1		Byte Order	Rage <dynamically set=""></dynamically>
e	EmpNo	integer	1	1		Ignore Case	🛃 no
e	Dept	integer	1	1		Fill Byte	퉜0
e	Empname	string	1	1		E Content	
	Address		1	1		Initiated Content	昦 no
	sequence		1	1		Sequence Kind	🛃 ordered
	StreetName	string	1	1		 Alignment 	
		string	1	1		 Delimiters 	
		sung	1	1			昂,
		string	1	1		Initiator	嚞 <no initiator=""></no>
e	Tel	<string></string>	1	1			搹 <no terminator=""></no>
e	Salary	decimal	1	1		Output New Line	🛃 %CR;%LF;
Add a Local Element							

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

3. You will change the elements in the Address global 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 global 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.

O CompanyAddressBin.xsd ⊠		_			
			A		
Test Parse Model Test Serialize Model Hide propertie	es Show bas	ic Show	all sections	Focus on se	elected S
•Schema 🛛 📮 📮 📮 📮 🖓	D 🗐	2: E	E, E, (È E	
Namespace <null namespace=""></null>					Chang
▼Schema References 🛛 🔉 💥					
A schema file in the same namespace uses an inclu	de. A schema	a file in a d	ifferent nam	espace uses	an import
Imports IBMdefined/RecordSeparatedFieldForm	nat.xsd ht	tp://www.	ibm.com/dfc	dl/RecordSep	aratedFie
A message is a global element that models an entire of	locument of d	lata.			
Name	Туре	Min Oc	cursk Max C	Decurs	
🖃 🖻 CompanyAddressBin					
🖃 🚥 sequence		1	1		
E CompanyName	string	1	1		
: 😑 🖻 Employee		1	unbou	inded	

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

A simple type defines the allowed values for one or more simple elements.	
---	--

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 Simp	ple Type	미뇌
Name:	TwoBytesBin	
Inherit from:	short	-
	OK Cancel	

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

🚯 Test Parse Model	🔓 Test Serialize Model	Hide properties	字 Show basic	A Hide empty sections	i Focus on selected
	🖃 🚥 sequence		1	1	
:	e EmpNo	inte	ger 1	1	
1	e Dept	inte	ger 1	1	
1	e Empname	stri	ng 1	1	
1	🖃 🖻 Address		1	1	
	🖃 🚥 sequenc	e	1	1	
-	e Stree	etName strii	ng 1	1	
1	e City	stri	ng 1	1	
1	e ZipC	ode strii	ng 1	1	
1	e Tel	<st< td=""><td>ring> 1</td><td>1</td><td></td></st<>	ring> 1	1	
1	e Salary	dec	imal 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 Test Serialize Model H	lide properties S	how advanced Show al	sections F	Focus on selected Show quick outline	create logical instance		
▼Schema 🛛 🛛 🖉 🖉 💹	2 J & M		6 h 🔶	Representation Properties (X)=	Variables		
				TwoBytesBin (Type)			
Namespace <nuii namespace=""></nuii>			5				
Schema References (0 include)	s, 1 import)			<type filter="" text=""></type>			3
A schema file in the same namespace	uses an include. A	schema file in a different	t namespa	Property	Value	?	
				Comment			
▼Messages 🔲 🖪 Ĥ 🖁		-		General			
		event of data		Encoding (code page)	🛃 <dynamically set=""></dynamically>		
A message is a global element that mode	is an enure docum	ent of data.		Byte Order	🛃 <dynamically set=""></dynamically>		
	T	0	2.6.1	Content	short		
Name	i ype Min	Occurs Max Occurs	Detaur	Representation	뿸 text		
🖃 e CompanyAddressBin				Length Kind	🛃 delimited		
🖃 🚥 sequence	1	1		Text Content			
e CompanyName	string 1	1		Text Number Representation	뿸 standard		
: e Employee	1	unbounded		Escape Scheme Reference	RecSepFieldsFmt:RecordEscapeSch		
Add a Local Element				Delimiters			
				Initiator	😤 <no initiator=""></no>		
▼Simple Types 🛛 📮 🗱				Terminator	📇 <no terminator=""></no>		
A simple type defines the allowed values	for one or more s	imple elements.		Validation	short		
Name Base Ty	pe						
TwoBytesBin short							
Data Formats (1 format)							
A data format is a container of DFDL pro	perties.			1			

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".

E Content	integer	•
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. And set the Binary Number Check Policy to "lax". ("Strict" will also work in this example). Remember – you need to "Show Advanced" for this property to be shown.

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

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.

0 Com	panyAddressBin.xsd 🛛					
Test Pa	rse Model Test Serialize Model Hide prope	rties Show	v basic Hide emp	🔺 ty sectio	ns Focus on selected Show quick outlin	e Create logical instance
★Mess A mess	ages 🚽 💭 🐙 🗘 🤸 🕱 🔤	Ei Ei document ol	f data.		Representation Properties	Asserts and Discriminators 🕺 Variables (n
Na	me	Туре	Min Occurs	Max C	<pre><type filter="" text=""></type></pre>	××
	🖃 🖻 CompanyAddressBin				Property	Value
	sequence		1	1	Comment S	Valac
-	e CompanyName	string	1	1	General	
1	😑 🖻 Employee		1	unbou	Data Format Reference	<default format=""></default>
	🖃 🚥 sequence		1	1	Encoding (code page)	<pre></pre>
1	e EmpNo	integer	1	1	Byte Order	<pre>dynamically set></pre>
	e Dept	integer	1	1	Ignore Case	見 no
	e Empname	string	1	1	Fill Byte	昱 0
- 8	🖃 🖻 Address		1	1	E Content	string
	🖃 🚥 sequence		1	1	Representation	R text
1	e StreetName	string	1	1	Length Kind	晃 delimited
	e City	string	1	1	The test of the test of the test of test	晃 false
	e ZipCode	string	1	1	Default Value S	<unset></unset>
:	e Tel	<string></string>	1	1	Fixed Value S	<unset></unset>
3	e Salary	decimal	1	1	🖃 Text Content	
Ad	d a Local Element				String Justification	🛃 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 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)

StreetName (Element)

<type filter="" text=""></type>				
Value	?			
<default format=""></default>				
🛃 <dynamically set=""></dynamically>				
🛃 <dynamically set=""></dynamically>				
🛃 no				
暑の				
string				
🛃 text				
prefixed				
뿸 characters				
TwoBytesBin				
yes				
no Kanser>				
<unset></unset>				
	Value Value <default format=""> <dynamically set=""> <</dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></dynamically></default>			

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).

Parse Mod	lel Test Serialize Model	III Hide propertie	s Show basic	Aide empty	sections	Focus on selected Sh	iow quick outline Create login
Message : message is	s いい いっぽう いっぽう いっぽう いっぽう いっぽう いっぽう いっぽう いっ	odels an entire d	locument of da	a.	_	Representation Pr	operties 📃 Asserts and I
Name			Туре	Min Occurs	Max (<type filter="" text=""></type>	
	e CompanyAddressBin						
	🖃 🚥 sequence			1	1	Property	Value
1	e CompanyName		string	1	1	- General	Defense adafadi famaka
	🖃 🖻 Employee			1	unbol	Data Format	Referenc <u><derault format=""></derault></u>
	🖃 🚥 sequence			1	1	Encoding (co	de page) 😤 <dynamically se<="" td=""></dynamically>
:	e EmpNo		integer	1	1	Byte Order	The second secon
- :	e Dept		integer	1	1	Fill Puto	
:	e Empnam	e	string	1	1	Content	62 V
:	🖃 💽 Address			1	1	Initiated Cor	itent 🗏 no
	🖃 🚥 sequ	ience		1	1	Sequence Kir	nd 🖳 ordered
0	e S	treetName	string	1	1	🕀 Alignment	
	eC	iity	string	1	1	Delimiters	
3	e Z	ipCode	string	1	1		,
1	e Tel		<string></string>	1	1	Initiator	晃 <no initiator=""></no>
- :	e Salary		decimal	1	1		
Add a Lo	cal Element					Output New	Line 🖳 %CR;%LF;

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.

0 CompanyAddr	essBin.xsd 🗙			
E	E		E E	
Test Parse Mode	The Control of the Co	- Mandal Channessee Channessee Channell and the	Contraction of the second seco	
Namespace	D Test Pars	e Model		
	- Message -			
<u>▶Schema</u>	Select mes	sage for testing. More		
A schema f	Message	File Selection		-
		Calast an innut film		
▼Messages	Parser In	Select an input file:		
A message is	Select cor	🖃 🔚 MessageModellingPrefixLib		
-	C Conte	IBM defined		
Name	Conte	Company Prefix Bin txt	ו•	
	Input file	Company, Prefix, Char, txt	gLength.tx 💌 Bro	wse
-				
	Specify run			
:	Runtime (
	Provide ru		be dynamically set. <u>Mo</u>	<u>re</u>
	Encoding			•
	Election r			
	r loading p			
•	Byte orde			
		Select an input file from the file system		
🔄 Navigator 🛛 🛃	Runtime			
OFDL Test - Pars	🔽 Valida		JW50	
Status: Parsing				
Input	Restore Def			
Data: /Messag		Ок (ancel	
-Parsed Input -	-			
Characters –				
1 Compa				

17. Close the yellow completion pop-up.



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

🗞 Navigator 🔝 Problems 🖺 DFDL Test - Parse 🕴 🕞 DFDL Test - Serialize 🕼 DFDL Test - Trace	🖻 i 🖪 🐚 🗏 🤣 📔
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: MessageModelingPrefixLb/Company.Prefix.Bin.txt	AddressBin.×sd)
Parsed Input	
Characters	
1 Company [compName=My Company	
2 Employee (empNum-1111111 dept-500 empName-Alice Wong Addr: 08200 Warden Avell "Markham, Ont" 013 H7 tel-90	5-347-5649 <mark> sal=</mark>
3 Employee (empNum=222222 dept=500 empName=James May Addr: M23 The Cuttings Chatham CH2 2PR tel=208-203-1	.332 sal= 189599.
4 Employee (empNum=333333 dept=310 empName=Richard Hammond Addr: 1016 Great Windmill London W2 3RJ tel=207	'-445-2955 <mark> sal=</mark> 5
5 Employee (empNum=444444 dept=230 empName=Jeremy Clarkeson Addr:N"Rose Cottage, Pea Dr"	
6 Gloucester GL01 2NM tel=743-123-4567 sal=75599.95)	
7 Employee (empNum=555555 dept=650 empName=Humphrey Littleton Addr: 0411 Regent Street ILondon INW1 1QT tel	=207-883-1238 s
8	
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.

🖬 DFDL Test - Logical Instanc	🖬 🛛 🖾 🗖 🗖				
Data source: <from 'dfdl="" -="" parse'="" test="" view=""></from>					
$\label{eq:message:companyAddressChar} MessageModellingPrefixLib/CompanyAverageModellingPrefixCib/$					
Tree View XML View					
Name	Туре	Value	▲		
CompanyAddressChar					
CompanyName	xs:string	My Company			
🖃 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 Binary scenario.

Version 9.0.0.0