

IBM Integration Bus

Message Modeling with DFDL

Lab 2 Modeling fixed-length data using a COBOL copybook

June 2015 Hands-on lab built at product Version 10.0.0.0

1.	INTRODUCTION	.3
2.	CREATING A MESSAGE MODEL FROM A COBOL COPYBOOK	.4
3.	TESTING THE MESSAGE MODEL	20
4.	USING THE TRACE FACILITY	29
END	OF LAB GUIDE	34

1. Introduction

In this lab, you will create a Message Model from a COBOL Copybook. Then you will test parse it against a valid data file and a malformed data file. In this last part you will be able to take a look at the trace facility.



This lab should be done after Lab 1, Message Modelling with CSV files.

2. Creating a Message Model from a COBOL Copybook

This lab shows you how to create a Message Model based on a fixed length COBOL Copybook format. To do that, you will use the message model wizard taking a .cpy file as input.

1. The first lab in this series created a CSV Message Model, so you should already have a Library created for this purpose. This lab uses the library called MessageModellingLibrary.

(If you didn't do that lab, create a new library now called MessageModellingLibrary).

In this library, click "<u>New</u>...", and select Message Model.



2. In the Message Model wizard, select COBOL and click Next.

🜐 New Message Model	
Create a new message r	nodel file
Select the message model ty	pe or format
XML	
SOAP XML	XML data for use in Web Services.
O Other XML	All other XML data.
Text and binary	
C CSV text	Comma Separated Values data, a delimited text format commonly used as an export format by spreadsheets and databases.
O Record-oriented text	Text data formats where delimited fields are grouped into records.
COBOL	Data for COBOL programs
О С	Data for C programs
O Other text or binary	All other text or binary data formats.
Enterprise Information Syst	tems
C SAP	Data from SAP systems including IDoc and BAPI
O Siebel	Data from Siebel systems
O PeopleSoft	Data from PeopleSoft
C JD Edwards	Data from JD Edwards systems
Other	
	Data from CORBA
O Database record	Records from relational databases
	Data for extended email format
O IBM supplied	Predefined data format
?	< Back Next > Finish Cancel

3. Leave the default option selected, and click Next.



4. Set the "Application or Library" to MessageModellingLibrary, by using the Browse button.

Select "Select source file from outside workspace". Click the Browse button.

🜐 New DFDL Schema	_ (2)
New Data Format Description Language Schema (DFDL) From a Source	
Select the type of data definition from which to import and create a DFDL schema.	
Application or Library: MessageModellingLibrary	Browse New
Folder: Folder Folder Folder Folder Folder	Browse
DFDL schema file name: <pre></pre>	
C Select source file from workspace:	
Herico MessageModelingLibrary	
€ Select source file from outside workspace:	
Location	Browse
☑ Overwrite existing DFDL schema file	
Copy source file into the 'importFiles' directory of the target project	
2	< Back Next > Einich Cancel
\lor	Carcel

5. Browse in C:\student10\MessageModeling\resources\ and select the file "PURCHASES.cpy".

Click Next.

🌐 New DFDL Schema							_ 8 ×
New Data Format De Select the type of data of	escription Language Schema (DFDL) From a definition from which to import and create a DFDL schem	a Source na.					
Application or Library: Folder:	MessageModelingLibrary <specifying a="" folder="" is="" optional=""></specifying>					Browse.	New
C Select source file from	n workspace:						
© Select serve fis for Locat in C: (studer ✓ Overwrite existing D ✓ Copy source file into	todelingLbrary todelingUbrary t10\MessageModeling\yesources\PURCHASES.cpy OL schema file the `mportFiles' directory of the target project					.	Browse
•			1	< Back	Next >	Finish	Cancel

6. Click on the ">>" button to select all found objects (just one in this case) and click Next.

Do not click Finish.

New DFDL Schema	
Structure and Message Selection Select the 01 level COBOL structures to import as messages.	S
PURCHASES	Imported structures
0	< Back Next > Finish Cancel

7. Leave most of the default values, but select "Recognize null values for all fields" and "Create value constraints from level 88 VALUE clauses".

Click Next.

🌐 New DFDL Schema				_ 8 ×
Import Options				
Optionally override defaults for generation of DFDL schema from COBOL import.				S
∏ Default values				
Specify these options if you want default values to be created for each field. Create default values from initial VALUEs				
_ Null values				
Specify this option if you want the processor to recognize fields as logically null when the value is SPACES, HIGH-VALUES or LOW-VALUES. Recognize null values for all fields	. <u>More</u>			
strings: SPACES v numbers: LOW-VALUES				•
Pad character				
Specify the character to be used by the processor when padding field values on output, and when trimming field values on input.				
⊂ Value constraints				
Spearly thill option if you want helds to be given value constraints whenever possible. More Creat value constraints from level 88 VALUE dauses				
- Field names				
Specify this option to stop upper-case COBOL field names being changed to lower-case. More				
V Preserve the case of held names				
Restore Defaults				
0	(Nm4 >	Circle 1	Grand
	< Bac	Next >	Finish	Cancel

8. Leave all the defaults values, and click Finish.

New DFDL Schema			_ 8 ×
mport Options Specify settings that describe the COBOL data as it appears on the target system.			s
The importer compiles the supplied COBOL file in order to create the DFDL schema. Some compiler options may be changed on this page. A full set of compiler options may be found in the Preferences settings here.			
Specifying target platform affects selection of the encoding options.			
Platform: Win32			•
Encoding options Organic - provided to the processor by the application at runtime Organic - provided halow			
Encoding (code page): ISO-8859-1			~
Floating point format: IEEE Non-Extended			
Byte order: 💿 Little endian 🔿 Big endian			
External (zoned) decimal Use EBCDIC sign characters with ASCII numbers			
Complete options			
QUOTE: @ DOUBLE C SINGLE			
TRUNC: © STD C OPT C BIN			
NSYMBOL: • NATIONAL C DECS			
Restore Defaults			
(3)	< Back Next	Finish	Cancel

9. The DFDL editor opens with the newly created DFDL message model called PURCHASES.xsd.

DURCH	PURCHASES.xsd 🛛							
Test Parse	Model Test Serialize Model Hide p	properties Show all sections	Focus on sele	cted Show c	ਿ quick outline	Create logical instance		
Messages	essages 🖉 🖓 🖑 🗱 🖬							
message is	nessage is a global element that models an entire document of data.					PURCHASES (Element)	(?)	
	8							
Name		Туре	Min Occurs	Max Occurs	Default Vi	Þ 🛱 🙀 🗶	🍇 🖽 📼 🖶	
- 6	PURCHASES	PURCHASES				Property Value	(?)	
Ģ] 🚥 sequence		1	1		Comment S		
4	e REQUEST_TYPE	<picx_string></picx_string>	1	1		General		
1	e RET_CODE	<picx_string></picx_string>	1	1		Data Format Re <u><default for<="" u=""></default></u>	<u>mat></u>	
1	e CustomerId	<picx_string></picx_string>	1	1		Encoding (code 🛃 <dynamic< td=""><td>cally set</td></dynamic<>	cally set	
1	e CustomerLastName	<picx_string></picx_string>	1	1		Byte Order 🛛 🛃 <dynamic< td=""><td>cally set</td></dynamic<>	cally set	
1	e CustomerFirstName	<picx_string></picx_string>	1	1		Ignore Case 🛛 🕌 yes		
1	e CustomerCompany	<picx_string></picx_string>	1	1		Fill Byte 🛃 %#r00;		
1	e CustomerAddr1	<picx_string></picx_string>	1	1		Content		
1	e CustomerAddr2	<picx_string></picx_string>	1	1		Length Kind implicit		
3	e CustomerCity	<picx_string></picx_string>	1	1		Occurrences		
1	e CustomerState	<picx_string></picx_string>	1	1		Min Occurs S 🔒 1		
1	e CustomerCountry	<picx string=""></picx>	1	1		Max Occurs S 😤 1		
1	e CustomerMailCode	<picx string=""></picx>	1	1		Alignment		
1	e CustomerPhone	<picx string=""></picx>	1	1		Delimiters		
1	e Customeri astUpdateDate	<picx_string></picx_string>	1	1		Initiator 😤 <no initia<="" td=""><td>tor></td></no>	tor>	
1	e PurchaseCount	<pic9-comp_short></pic9-comp_short>	1	1	0	lerminator 😤 <no td="" term<=""><td>inator></td></no>	inator>	
	+ Purchase	di teo comp_non n	0		-	Cutput Now Line 0 CD 9/1	F .	
1		<picx_string></picx_string>	1	1			F;	
Addalo	cal Element	Stitun_outrige	1	1				
	<u>di cicilerie</u>					//		

10. Notice the wizard automatically added a file called "CobolDataDefinitionFormat.xsd" under the Schema Definitions in MessageModellingLibrary.

This file is referenced by PURCHASES.xsd as a schema import, and it contains COBOL-specific defaults for all the DFDL properties, and some pre-defined simple types.



11. Expand the "Other Resources" folder under the MessageModelingLibrary library.

Expand the "importFiles" folder and you will see the PURCHASES.cpy file that the wizard has automatically imported.



12. Double-click PURCHASES.cpy to open it in the editor.

DURCHASES.xsd	T PURCHASES.cpy 🛛	
+	2+3+4	-+5+6+
01 PURC	HASES.	
03	REQUEST-TYPE	PIC X.
03	RET-CODE	PIC XX.
03	CustomerId	PIC X(8).
03	CustomerLastName	PIC X(20).
03	CustomerFirstName	PIC X(20).
03	CustomerCompany	PIC X(30).
03	CustomerAddr1	PIC X(30).
03	CustomerAddr2	PIC X(30).
03	CustomerCity	PIC X(20).
03	CustomerState	PIC X(20).
03	CustomerCountry	PIC X(30).
03	CustomerMailCode	PIC X(20).
03	CustomerPhone	PIC X(20).
03	CustomerLastUpdateDate	PIC X(8).
03	PurchaseCount	PIC 9(3) USAGE COMP.
03	Purchase OCCURS 0 TO 99 TIME	S
	DEPENDING ON PurchaseCount.	
04	PurchaseId	PIC 9(5).
04	ProductName	PIC X(30).
04	Amount	PIC 9(2).

This is a simple copybook with:

- 1. 14 string fields
- 2. PurchaseCount: binary field with the number of the Purchase structure occurrences
- 3. Purchase: Repeating structure
 - 1. Purchaseld, Amount: numeric fields.
 - 2. Price: numeric field with 2 decimal places.

Message Modelling - Fixed-length using COBOL

13. Switch to the DFDL editor. For the PURCHASES.xsd This shows the string fields, defined as "PICX_string" by the import wizard:

Name	Туре	Min Occurs	Max Occurs
	PURCHASES		
🖃 🚥 sequence		1	1
E REQUEST_TYPE	<picxstring></picxstring>	1	1
E RET_CODE	<picxstring></picxstring>	1	1
E CustomerId	<picxstring></picxstring>	1	1
E CustomerLastName	<picxstring></picxstring>	1	1
E CustomerFirstName	<picxstring></picxstring>	1	1
E CustomerCompany	<picxstring></picxstring>	1	1
E CustomerAddr1	<picxstring></picxstring>	1	1
E CustomerAddr2	<picxstring></picxstring>	1	1
E CustomerCity	<picxstring></picxstring>	1	1
E CustomerState	<picxstring></picxstring>	1	1
E CustomerCountry	<picxstring></picxstring>	1	1
E CustomerMailCode	<picxstring></picxstring>	1	1
E CustomerPhone	<picxstring></picxstring>	1	1
E CustomerLastUpdateDate	<picxstring></picxstring>	1	1
E PurchaseCount	<pic9-compshort></pic9-compshort>	1	1
🗄 🕀 🖻 Purchase		0	99
E RETURN_COMMENT	<picxstring></picxstring>	1	1

Add a Local Element

14. In the DFDL Editor click on the "CustomerLastName" field to see its properties:

est Parse Model Test Serialize Model H	de properties Show all sections	Focus on sele	ected Show qu	ick outline Create logical instance			
essages 🛛 💂 🛊 🦆 🌡	(🖡 🖬			Representation Properties	XI = Variables E: Asserts and Discriminators		
essage is a global element that models an	entire document of data.			CustomerLastName (Element)	0		
Name Type Min Occurs Max Occurs <type filter="" text=""></type>							
e PURCHASES	PURCHASES			Property	Value (?)		
🖃 🚥 sequence		1	1	Comment S	PIC X(20) display		
e REQUEST_TYPE	<picx_string></picx_string>	1	1	🖃 General			
e RET_CODE	<picx_string></picx_string>	1	1	Data Format Reference	<default format=""></default>		
e CustomerId	<picx_string></picx_string>	1	1	Encoding (code page)	R <dynamically set=""></dynamically>		
e CustomerLastName	<pre><picx_string></picx_string></pre>	1	1	Byte Order	昂 <dynamically set=""></dynamically>		
e CustomerFirstName	<picx_string></picx_string>	1	1	Ignore Case	Pa yes		
e CustomerCompany	<picx_string></picx_string>	1	1	Fill Byte	屠, %#r00;		
e CustomerAddr1	<picx_string></picx_string>	1	1	Content	string		
e CustomerAddr2	<picx_string></picx_string>	1	1	Representation	Ha text		
e CustomerCity	<picx_string></picx_string>	1	1	Length Kind	দ explicit		
e CustomerState	<picx_string></picx_string>	1	1	Length	20		
e CustomerCountry	<picx_string></picx_string>	1	1	Length Units	품a bytes		
e CustomerMailCode	<picx_string></picx_string>	1	1		une		
e CustomerPhone	<picx_string></picx_string>	1	1	Eived Value	(incet)		
e CustomerLastUpdateDa	e <picx_string></picx_string>	1	1	Text Content	(unset)		
e PurchaseCount	<pic9-comp_short></pic9-comp_short>	1	1	String Justification	E left		
E Purchase		0	99	String Pad Character	层 %SP:		
e RETURN_COMMENT	<picx_string></picx_string>	1	1	Truncate Specified Length	Str 🗏 no		
Add a Local Element				Pad Kind	P. padChar		
				Trim Kind	見 padChar		
				Escape Scheme Reference			

In the properties view, look for the "Content" section. Note that the field was modeled as "text" representation, with a fixed (explicit) Length of 20 bytes, because the cpy file defined it as a "PIC X(20)"

0] PURCHASES.xsd 🕅 🗹 PURCHASES.cpy								
Esk Englise Model Hide properties Show all sections Focus on selected Show quick outline Create logical instance								
Asserts and Discriminators								
iessage is a global element that models an entire document of data.				PurchaseCount (Element)	0			
Name	Туре	Min Occurs	Max Occurs	<type filter="" text=""></type>	📑 🛃 🗶 💥 🖽 🕞 💠			
PURCHASES	PURCHASES			Property	Value (?)			
see sequence		1	1	Comment 🛐	PIC 9(3) binary			
E REQUEST_TYPE	<picx_string></picx_string>	1	1	🖃 General				
E RET_CODE	<picx_string></picx_string>	1	1	Data Format Reference	<default format=""></default>			
e CustomerId	<picx_string></picx_string>	1	1	Encoding (code page)	棍 <dynamically set=""></dynamically>			
e CustomerLastName	<picx_string></picx_string>	1	1	Byte Order	🛃 <dynamically set=""></dynamically>			
e CustomerFirstName	<picx_string></picx_string>	1	1	Ignore Case	界 yes			
e CustomerCompany	<picx_string></picx_string>	1	1	Fill Byte	冕 %#r00;			
e CustomerAddr1	<picx string=""></picx>	1	1	Content	unsignedShort			
e CustomerAddr2	<picx string=""></picx>	1	1	Representation	뮴a binary			
e CustomerCity	<picx_string></picx_string>	1	1	Length Kind	🛃 explicit 📃			
e CustomerState	<picx_string></picx_string>	1	1	Length	📮 2			
	<picx_string></picx_string>	1	1	Length Units	Pa bytes			
e CustomerMailCode	<picy_string></picy_string>	1	1	+ Nillable 5	true			
	<picx_string></picx_string>	-	1	Default Value S	0			
Customer Priorie	<picx_string></picx_string>	1	1	Fixed Value S	<unset></unset>			
: CustomerLastupdateDate	<pica_string></pica_string>	1	1	Binary Content				
	<pic9-comp_snort></pic9-comp_snort>	1	1	Number Check Policy	男 lax			
: 🗄 e Purchase		U	99	Number Representation	昂 binary			
e RETURN_COMMENT	<picx_string></picx_string>	1	1	Occurrences				
Add a Local Element				Min Occurs S	累 1			

15. In the DFDL Editor, click on the PurchaseCount field to see its properties:

This field, which was defined as binary in the copybook file ("PIC 9(3) USAGE COMP"), was created as "PIC9_Comp_short" by the Import wizard.

You can see the details of this field in the properties view, where its length is set to "2", its Length Units to "bytes" and its representation to "binary".

Also, in the "Binary Content" section, its Binary Number Representation is set to binary. This property can take 4 different values:

- packed: represented as a packed decimal. Each byte contains 2 decimal digits except for the least significant byte, which contains a sign in the least significant nibble
- bcd: represented as a binary coded decimal with 2 digits per byte.
- binary: represented as 2' complement for signed types and unsigned binary for unsigned types.
- ibm4690Packed: used by the IBM 4690 retail store devices

16. Now click on the Price field, in the Purchase structure.

: 😑 🖻 Purchase		0	99	
🖃 🚥 sequence		1	1	
e PurchaseId	<pic9-display-zonedint></pic9-display-zonedint>	1	1	
e ProductName	<picxstring></picxstring>	1	1	
e Amount	<pic9-display-zonedshort></pic9-display-zonedshort>	1	1	
: e Price	<pic9-display-zoneddecimal></pic9-display-zoneddecimal>	1	1	
E RETURN_COMMENT	<picxstring></picxstring>	1	1	

17. In the properties view, look at the "Content" section.

🔲 Representation Properties 🛛 🗱 Variables 📄 Asserts and Discriminators				
Price (Element)				
<type filter="" text=""></type>	🔜 🕌 🔜 🗶 💥 🗈 🖨 🖶			
Property	Value 🕜 🔺			
Comment S	PIC 9(8)V9(2) display			
General				
Data Format Reference	<default format=""></default>			
Encoding (code page)	R <dynamically set=""></dynamically>			
Byte Order	Ra <dynamically set=""></dynamically>			
Ignore Case	류 yes			
Fill Byte	昂 %#r00;			
Content	decimal			
Representation	昂 text			
🖃 Length Kind	掃 explicit			
Length	10			
Length Units	掃 bytes			
+ Milable S	TUE			
Default Value 💲	0			
Fixed Value S	<unset></unset>			
Text Content				
Decimal Signed	no			
	扂 zoned			
Number Justification	昂 right			
Number Pad Character	昂 0			
Pad Kind	💂 padChar			
Trim Kind	🗏 padChar			
Sample Test Data				

Note that it is defined as a decimal field, with text representation and a length of 10 bytes (8 integers and 2 decimal places).

18. Look at the "Text Content" section of the properties view.

Representation Properties (X)= Variables E Asserts and Discriminators						
Price (Element)	Price (Element)					
<type filter="" text=""></type>	<type filter="" text=""></type>					
Property	Value (?)					
Text Content						
Decimal Signed	no					
 Number Representation 	扂 zoned					
Number Check Policy	튐 lax					
Number Pattern	0000000000+					
Rounding	<u> 봄</u> pattern					
Sign Style	🔁 asciiStandard					
Number Justification	🖥 right					
Number Pad Character						
Pad Kind	PadChar					
Trim Kind	PadChar					
Escape Scheme Reference	R <u><no escape="" scheme=""></no></u>					

Note that the "Number Representation" is defined as "zoned", with a pattern of 8 integer numbers and 2 decimal places.

The letter "V" in the Number Pattern is an implied decimal point (common in COBOL copybooks).

19. Click on the "Purchase" element to open its properties.

st Parse Model Test Serialize Model Hide pr	pperties Show all sections Focus or	selected Sho	ti w quick out	ine Create logical instance		
ame	Туре	Min Occurs	Max C	Representation Properties	Asserts and Discriminators	
e PURCHASES	PURCHASES			Purchase (Element)		C
sequence		1	1			
e REQUEST_TYPE	<picxstring></picxstring>	1	1	<type filter="" text=""></type>	📑 🛃 🗙 🔆 🖽	
e RET_CODE	<picxstring></picxstring>	1	1	Property	Value	(?)
e CustomerId	<picx_string></picx_string>	1	1	Comment S		
e CustomerLastName	<picxstring></picxstring>	1	1	🖃 General		
e CustomerFirstName	<picxstring></picxstring>	1	1	Data Format Reference	<pre><default format=""></default></pre>	
e CustomerCompany	<picx_string></picx_string>	1	1	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
e CustomerAddr1	<picx string=""></picx>	1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
e CustomerAddr2	<picx string=""></picx>	1	1	Ignore Case	暑 yes	
e CustomerCity	<picx string=""></picx>	1	1	Fill Byte	暑 %#r00;	
e CustomerState	<picx string=""></picx>	1	1	Content		
CustomerCountry	<picx string=""></picx>	1	1	Length Kind	implicit	
CustomerMailCode	<picx_string></picx_string>	1	1	Occurrences		
	<picy_string></picy_string>	-	-	Min Occurs S	0	
Customer astilladateDate	<picxstring></picxstring>	-	÷	Max Occurs S	99	
E Customercastopuatebate	<pica_sung></pica_sung>		1	Occurs Count Kind	expression	
	<pic9-comp_short></pic9-comp_short>	1	1	Floating	뤎 no	
		0	99	Alignment		
sequence		1	1	Delimiters		
e PurchaseId	<pic9-display-zonedint></pic9-display-zonedint>	1	1	Initiator	器 <no initiator=""></no>	
e ProductName	<picx_string></picx_string>	1	1	Terminator	點 <no terminator=""></no>	
e Amount	<pic9-display-zonedshort></pic9-display-zonedshort>	1	1	Empty Value Delimiter Policy	<u> </u>	_
e Price	<pic9-display-zoneddecimal></pic9-display-zoneddecimal>	1	1	Output New Line	點 %CR;%LF;	
e RETURN_COMMENT	<picx_string></picx_string>	1	1			

20. Look for the "Occurrences" section inside the properties view, and expand the "Occurs Count Kind" property.

PURCHASES.xsd X PURCHASES.cpy						- 8
Test Parse Model Test Serialize Model Hide pr	operties Show all sections Focus o	n selected Show	법 w quick out	ine Create logical instance		
Name	Type	Min Occurs	Max C	Representation Properties	Asserts and Discriminators	
	PURCHASES			Purchase (Element)		?
		1	1	Turenuse (cientency		
REQUEST TYPE	<picx string=""></picx>	1	1	<type filter="" text=""></type>	📑 🛃 🗙 💥 🖽	E 🖶
e RET_CODE	<picx_string></picx_string>	1	1	Property	Value	(?)
E CustomerId	<picx_string></picx_string>	1	1	Comment S		
e CustomerLastName	<picx_string></picx_string>	1	1	General		
E CustomerFirstName	<picx_string></picx_string>	1	1	Data Format Reference	<default format=""></default>	
e CustomerCompany	<picx_string></picx_string>	1	1	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
E CustomerAddr 1	<picxstring></picxstring>	1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
e CustomerAddr2	<picxstring></picxstring>	1	1	Ignore Case	昇 yes	
: e CustomerCity	<picxstring></picxstring>	1	1	Fill Byte	暑 %#r00;	
e CustomerState	<picx_string></picx_string>	1	1	Content		
: e CustomerCountry	<picxstring></picxstring>	1	1	Length Kind	implicit	
e CustomerMailCode	<picx_string></picx_string>	1	1	Occurrences	0	
: e CustomerPhone	<picxstring></picxstring>	1	1	Min Occurs S	0	
E CustomerLastUpdateDate	<picx_string></picx_string>	1	1		avoression	
e PurchaseCount	<pic9-compshort></pic9-compshort>	1	1		{ /PurchaseCount}	
E Purchase		0	99	Floating		
🖃 🚥 sequence		1	1	Alignment		
: e PurchaseId	<pic9-display-zonedint></pic9-display-zonedint>	1	1	Delimiters		
: e ProductName	<picxstring></picxstring>	1	1	Initiator	晃 <no initiator=""></no>	
: e Amount	<pic9-display-zoned_short></pic9-display-zoned_short>	1	1	Terminator	暑 <no terminator=""></no>	
: e Price	<pic9-display-zoneddecimal></pic9-display-zoneddecimal>	1	1	Empty Value Delimiter Policy	掃 none	
E RETURN_COMMENT	<picxstring></picxstring>	1	1	Output New Line	暑 %CR;%LF;	
Add a Local Element			-			
▲			Þ	Sample Test Data		

This property, as defined by the DFDL specification, can take different values:

- 1. fixed: uses the "maxOccurs" property
- 2. expression: uses the value defined by the expression in "occursCount" property.
- 3. parsed: the number of occurrences is determined by normal speculative parsing.
- 4. implicit: uses "minOccurs" and "maxOccurs" properties with speculative parsing

In this case, the "OccursCountKind" property is set to "expression", and "occursCount" is set to point to the "PurchaseCount" element. This means that the number of occurrences of the "Purchase" repeating structure will be defined by the PurchaseCount element.

This was defined by the Import wizard to reflect the cpy file, which stated:

Purchase OCCURS 0 TO 99 TIMES DEPENDING ON PurchaseCount.

Notice also that the MinOccurs property is set to "0" and the MaxOccurs property is set to "99", as the cpy file stated.

21. Save your message model (PURCHASES.xsd) by pressing Ctrl+S, or File->Save.

3. Testing the Message Model

1. Now you will test that the message model correctly models the COBOL data. Click the "Test Parse Model" icon.

DURCHASES.xsd 🔀 🗹 PURCHASES.cpy						- 8
Ei Ei Test Parse Model Hide pr	operties Show all sections Focus or	selected St	ti now quick out	line Create logical instance		
Name	Туре	Min Occur	s Max C	Representation Properties	Asserts and Discriminators	
	PURCHASES			Purchase (Element)		?
sequence		1	1		6 - 3 - 3	
e REQUEST_TYPE	<picx_string></picx_string>	1	1	<type filter="" text=""></type>	📑 🛃 🗶 🔆 🖽	E 🖶
e RET_CODE	<picx_string></picx_string>	1	1	Property	Value	(?)
e CustomerId	<picx_string></picx_string>	1	1	Comment S		
e CustomerLastName	<picx_string></picx_string>	1	1	General		
E CustomerFirstName	<picx_string></picx_string>	1	1	Data Format Reference	<default format=""></default>	
E CustomerCompany	<picx_string></picx_string>	1	1	Encoding (code page)	🛃 <dynamically set=""></dynamically>	
E CustomerAddr 1	<picx_string></picx_string>	1	1	Byte Order	🛃 <dynamically set=""></dynamically>	
E CustomerAddr2	<picx_string></picx_string>	1	1	Ignore Case	昇 yes	
e CustomerCity	<picx string=""></picx>	1	1	Fill Byte	暑 %#r00;	
e CustomerState	<picx _string=""></picx>	1	1	Content		
e CustomerCountry	<picx string=""></picx>	1	1	Length Kind	implicit	
e CustomerMailCode	<picx _string=""></picx>	1	1	Occurrences		
e CustomerPhone	<picx string=""></picx>	1	1	Min Occurs S	0	-
e CustomerLastUpdateDate	<picx string=""></picx>	1	1	Max Occurs S	99	
e PurchaseCount	<pic9-comp_short></pic9-comp_short>	1	1	Occurs Count Kind	expression	+
= e Purchase		0	99	Occurs Count	{/PurchaseCount}	
□ ···· sequence		1	1	Floating	器 no	
e PurchaseId	<pic9-display-zoned int=""></pic9-display-zoned>	1	1	+ Alignment		
e ProductName	<picx string=""></picx>	1	1	Initiator	con initiator >	
E Amount	<pic9-display-zoned_short></pic9-display-zoned_short>	1	1	Terminator		
e Price	<pic9-display-zoned_decimal></pic9-display-zoned_decimal>	1	1	Empty Value Delimiter Policy		
	<picy_string></picy_string>	1	1	Output New Line		
Add a Local Element	<pica_joing></pica_joing>		· _	ouguertan and		
			▼ 	Sample Test Data		

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

Test Parse Mode	2			
Message				,
Select message for	r testing. <u>More</u>			
Message name:*	PURCHASES			•
Parser Input				
Select content to b	e parsed against schema.			
C Content from '	DFDL Test - Serialize' view			
Content from a	a data file			_
Input file name:* ⁶)		•	Browse
Specify runtime conf	figuration.			
-Runtime encoding	options			
Provide runtime va	lues for properties which h	ave been configured i	in the model to be dynamically	set. <u>More</u>
Encoding (code pa	ge): UTF-8			•
Eleating point form	at: IEEE Non-Extended			•
riousing point form				
Byte order: 🔿 Lit	tle endian 💿 Big endian			
Runtime validation				
Validate data a	gainst schema More			
estore Defaults				

3. In the File Selection dialog, select the "Select an input file from the file system" option.

Click on the Browse button.

File Selection		<u>_ 🗆 ×</u>
Select an input file:		
🖻 🕖 MessageModellingLibra	ary	
IBMdefined		
E Dog		
library.descriptor		
Select an input file from the	e file system	
		Browse
?	OK	Cancel

4. Navigate to "C:\student10\MessageModeling\data\" and select the "purchases.dat" file.

Click the Open button.

🌐 Open	×
G → ↓ + student 10 + MessageModeling + data	👻 🔽 Search data
Organize 🔻 New folder	III 👻 🗔 🔞
Image: Second Secon	Name ^ Companies - lots.txt Companies.txt Companies_lots_unordered.txt Company.txt employees.csv employees.WithoutCRLF.csv employees.NoHeader.csv purchases.BadPhoneNumber.dat purchases_MALFORMED.dat v Image: State of the system

5. Click OK on both windows.

🌐 Test Parse Model			_0×
Message Select message for Message name:*	File Selection Select an input file:	X	
Parser Input Select content to be Content from 'DI Content from a Input file name:* ⁰	MessageModellingLibrary		Browse
Runtime encoding o Provide runtime valu Encoding (code page Floating point forma Byte order: O Littl	Select an input file from the file syst	em	cally set. <u>More</u>
Runtime validation	C:\student10\MessageModeling\da	ta\purchase Browse	
Restore Defaults	?	OK Cancel	
		OK	Cancel

6. Click on the checkbox "Remember my decision", and click Yes.



7. The DFDL Test perspective will open, with the Test Parse view in focus.

A message balloon will appear, indicating the parsing was successful.

Close it by clicking on the "x", or by clicking anywhere else in the workbench.

1 PURCHASES.xsd 🖾 🗹 PURCHASES.cpy					🖬 DFDL Test - Logical Instanc	e X	🔓 🛛 🗳 🗖
Test Parse Model Test Serialize Model Show pro	operties Show all sections Focus or	selected S	Handre 1	•	Data source: <from 'dfdl<br="">Message: PURCHASES (/Us</from>	Test - Parse' viev ers/iibadmin/IBM,	w> /IIBT 10/workspace/Message
Name	Туре	Min Occur	s Max Occurs Default	Va			
e PURCHASES	PURCHASES				Tree View XML View		
a e sequence		1	1		Name	Type	Value
e REQUEST_TYPE	<picx_string></picx_string>	1	1		E PURCHASES	11900	- Vulue
E RET CODE	<picx string=""></picx>	1	1		REQUEST_TYPE	xs:string	A
e CustomerId	(PICX string)	1	1		RET_CODE	xs:string	00
	orrow allies	-	-	- 11	CustomerId	xs:string	12345678
: e CustomerLastName	<picxstring></picxstring>	1	1	_	CustomerLastName	xs:string	Griffin
e CustomerFirstName	<picxstring></picxstring>	1	1		CustomerFirstName	xs:string	Peter
E CustomerCompany	<picx_string></picx_string>	1	1		CustomerCompany	xs:string	Pawtucket Brew
E CustomerAddr1	<picx string=""></picx>	1	1		CustomerAddr1	xs:string	31 Spooner st.
E CustomerAddr?		1	1		CustomerAddr2	xs:string	456 1st av.
	Griex_sungs			, ČI	CustomerCity	xs:string	Quahoq
R: Navigator Problems L DFDL Ter Problems DFDL Ter Problems DFDL parser Status: Parsing completed: Wed Mar 11 1 Input - Sele Data: C:\student10\WessageModelIng\d -Parsed Input - To Characters - 1 A0012345678Griffin	arsing completed successfully. ecting an element in the DFDL editor w is view menu on the view toolbar provi view the logical instance that was crea view the trace captured while running o not display this message again	vill cause the p des options to ated by the D the DFDL par	parsed input to focus only o control how the data is FDL parser, dick the Ope rser, dick the Open DFDL	y on data displaye en DFDL I . Trace Vi	a pertaining to the selected ele d in the view. Click the arrow ic Logical Instance View toolabar bu iew toolbar button, or click <u>here</u>	ment. on on the toolba utton, or click <u>her</u> 3.	r or <u>here</u> to open the menu. .e.
Selected: PURCHASES Repeatin	ig index: 1 Range in parsed	input: 0 - 499	Character Selection : Row: 0 Column: 0	In Input	Byte Selection In Input	0	

8. Inspect the "Test - Logical Instance" view. Navigate through the message tree parsed from the input file.

💷 DFDL Test - Logical Instance	= XX				
Data source: <from 'dfdl="" 1<="" td=""><td colspan="5">Data source: <from 'dfdl="" -="" parse'="" test="" view=""></from></td></from>	Data source: <from 'dfdl="" -="" parse'="" test="" view=""></from>				
Message: PURCHASES (/Use	Message: PURCHASES (/Users/iibadmin/IBM/IIBT10/workspace/MessageModellingLibrary/PURCHASES.xsd)				
_	,,		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
Trac View MML View					
News	Turne	Value			
	Туре	value			
	verstring	٨			
	xsisuing	A 00			
REI_CODE	xsisting	10245679			
Customeria	xs:string	12345078			
CustomerLastName	xs:string	Grimn			
CustomerFirstvame	xs:string	Peter			
CustomerCompany	xs:string	Pawtucket Brew			
CustomerAddr1	xs:string	31 Spooner st.			
CustomerAddr2	xs:string	456 1st av.			
CustomerCity	xs:string	Quahog			
CustomerState	xs:string	Rhode Island			
CustomerCountry	xs:string	USA			
CustomerMailCode	xs:string	12312			
CustomerPhone	xs:string	123-123-1234			
CustomerLastUpdateD	xs:string	04082008			
PurchaseCount	xs:unsignedShort	4			
Purchase					
PurchaseId	xs:unsignedInt	1			
ProductName	xs:string	Beer			
Amount	xs:unsignedShort	6			
Price	xs:decimal	10.30			
Purchase					
PurchaseId	xs:unsignedInt	2			
ProductName	xs:string	Chips			
Amount	xs:unsignedShort	1			
Price	xs:decimal	2.25			
Purchase					
Purchase					
RETURN_COMMENT	xs:string	none			
	-				

Note that the parser shows "10.30" (2 decimal places) because the COBOL field was defined as PIC 9(8)V99.

9. In the DFDL Editor, click on any element on the Message Model and you will see the relevant data underlined in the input text below:

DURCHASES.xsd 🖾 🗹 PURCHASES.cpy		DFDL Test - Logical Instance	23		F E -
Test Parse Model Test Serialize Model Show or	operties Show all sections 3	Data source: <from 'dfdl="" t<="" td=""><td>'est - Parse' viev</td><td>w></td><td></td></from>	'est - Parse' viev	w>	
		Message: PURCHASES (/Use	rs/iibadmin/IBM,	/IIBT 10/workspace/Messag	geModellingLibrary/PURCHASES.xsd
Name	Type	a			
e PURCHASES	PURCHASES	Tree View XML View			22
sequence	1	Name	Туре	Value	
e REQUEST_TYPE	<picx_string> 1</picx_string>	PURCHASES			
e RET_CODE	<picx_string></picx_string>	REQUEST_TYPE	xs:string	A	
e CustomerId	<picx string=""></picx>	RET_CODE	xs:string	00	
	artex_stings	CustomerId	xs:string	12345678	
: e CustomerLastivame	<picx_string></picx_string>	CustomerLastName	xs:string	Griffin	
Contenes Firstillens	-OICX_strings	CustomerFirstName	xs:string	Peter	
E CustomerCompany	<picx_string></picx_string>	CustomerCompany	xs:string	Pawtucket Brew	
e Customer Addr 1	CPICA_SUNG2	CustomerAddr1	xs:string	31 Spooner st.	
: e CustomerAddr2	<picy string=""></picy>	CustomerAddr2	xs:string	456 1st av.	
	strex_stings	CustomerCity	xs:string	Quahog	
: e CustomerCity	<picx_string></picx_string>	CustomerState	xs:string	Rhode Island	
e CustomerState	<picx_string></picx_string>	CustomerCountry	xs:string	USA	
e CustomerCountry	<picx_string> 1_</picx_string>	CustomerMailCode	xs:string	12312	
- · ·		CustomerPhone	xs:string	123-123-1234	
		CustomerLastUpdateD	xs:string	04082008	
🕾 - Navigator 🖹 Problems 🕒 DFDL Test - Pars	se 🔀 🔚 DFDL Test - Serialize 🗈 DFDL	Test - Trace		0 🖣 🖬 🕞	i 🤣 🔢 🗵 🔍 🗖
DFDL Test - Parse: Runs the DFDL parser with the	e provided physical input data and selected	message, and updates the logical in:	stance view with	n the result of the parse.	
Status: Parsing completed: Wed Mar 11 11:28:16	5 GMT 2015				
- Input					
Data: C:\student10\MessageModeling\data\pur	chases.dat 💌 Browse Encoding	(code page): UTF-8	•	Message: PURCHAS	ES (/MessageModellingLibrary/PURG
Parsed Input					
Characters					
1 A0012345678Griffin	Peter	Pawtucket Brew	əry	11 Spoc	ner st.
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓					1

🚺 PURCHASES.xsd 🖾 🗾 PURCHASES.cpy		- C Ej	DFDL Test - Logical Instanc	e 23		6 G.	- 8	
Test Parse Model Test Serialize Model Show pr	operties Show all sections 3	- D	ata source: <from 'dfdl<="" td=""><td>Test - Parse' view></td><td></td><td></td><td></td></from>	Test - Parse' view>				
e CustomerId	<picx_string></picx_string>	I M	lessage: PURCHASES (/Us	ers/iibadmin/IBM/II	BT 10/workspace/Messa	geModellingLibrary/PURCHASE	S.xsd)	
e CustomerLastName	<picx string=""></picx>	1						
E CustomerFirstName	<picx_string></picx_string>	: 17	ee View XML View					
E CustomerCompany	<picxstring></picxstring>	1	lame	Type	Value		A A	
E CustomerAddr1	<picx_string></picx_string>	: .	- PURCHASES	1.77				
e CustomerAddr2	<picx string=""></picx>	1	REQUEST_TYPE	xs:string	A			
e CustomerCity	<picy string=""></picy>		RET_CODE	xs:string	00			
E Customer City	<picx_sung></picx_sung>		CustomerId	xs:string	12345678			
: e CustomerState	<picxstring></picxstring>		CustomerLastName	xs:string	Griffin			
e CustomerCountry	<picxstring></picxstring>	1	CustomerFirstName	xs:string	Peter			
e CustomerMailCode	<picx string=""></picx>	1	CustomerCompany	xs:string	Pawtucket Brew			
: CustomerPhone	<picx string=""></picx>	,	CustomerAddr1	xs:string	31 Spooner st.			
	erev i i i		CustomerAddr2	xs:string	456 1st av.			
: e CustomerLastUpdateDate	<picxstring></picxstring>		CustomerCity	xs:string	Quahog			
E PurchaseCount	<pic9-compshort></pic9-compshort>	1	CustomerState	xs:string	Rhode Island			
E Purchase		c	CustomerCountry	xs:string	USA			
			CustomerMailCode	xs:string	12312			
			CustomerPhone	xs:string	123-123-1234			
			CustomerLastUpdate	C xs:string	04082008		-	
🔁 Navigator 🚼 Problems 🕒 DFDL Test - Par	se 🖾 🛅 DFDL Test - Serialize	e 🕞 DFDL Test -	Trace		🕽 🖻 i 🖻 i 🕞 (a. 🤣 🔚 📧 💷 💎	- 0	
Input								
Data: C:\student10\MessageModeling\data\pur	chases.dat 💌 Browse	Encoding (code	page): UTF-8	•	Message: PURCHA	SES (/MessageModellingLibrary,	/PURC	
Parsed Input								
Characters								
1 -123-1234 040820	1 -123-1234 040820081000018eer 0600000103000002Chips							
Selection in DFDL Editor			Char	acter Selection In I	nput Byte Selection	In Input		
<u>1</u>								

10. Now click on the PurchaseCount element.

Since it's a binary field, the highlighted value isn't readable with this editor.

11. Now click the "Show hex" button (top right of the lower pane, as highlighted below).

😼 Navigator 🔀 Problems 🗳 DFDL Test - Parse 🔀 🔄 DFDL Test - Serialize 🕼 DFDL Test - Trace	U 🖣 🖬 🕞 🖳 🤣 🔛 🕇 🦳 🗸
Input	
Data: C:\student10\MessageModeling\data\purchases.dat 💌 Browse Encoding (code page):	UTF-8 Message: PURCHASES (/MessageModellingLibrary/PURC
- Darred Toput	
1 122 1224 04002000 0000 1P+++	
1 -123-1234 04062006 <u>11</u> 00001Beer	<u>00 04 </u> 0 30 30 30 31 42 65 65 72 20 20 20 20 20 7
·	
K P	
Selection in DFDL Editor	Character Selection In Input Byte Selection In Input
4	Þ

Notice that the binary field is now readable, and has a value of "00 04" which corresponds to the 4 occurrences of the "purchase" element.

Click the "Show Hex" to revert to normal display.

4. Using the Trace facility

1. Next you are going to test the message model using a malformed message.

In the DFDL Test perspective, "DFDL Test - parse" view, click on the Browse button.

🔁 Navigator 🔝 Problems 🕒 DFDL Tes	t - Parse 🛛 🛅 DFDL Test - Serialize 🕻	DFDL Test - Trace	0) 🕒 🖬 🕼 🛄 🤣 🔢 🗵 🗆 🗆
DFDL Test - Parse: Runs the DFDL parser v	with the provided physical input data and se	elected message, and u	updates the logical instance view with th	e result of the parse.
Status: Parsing completed: Wed Mar 11 1	1:28:16 GMT 2015			
Input Data: C:\student10\MessageModeling\da	ata\purchases.dat 💌 Browse Ei	incoding (code page):	UTF-8	Message: PURCHASES (/MessageModellingLibrary/PURCHA
Parsed Input				
Characters				
1 3-123-1234 0	4082008 <u>11</u> 00001Beer		0600000010300	00002Chips
Selection in DFDL Editor Selected: PurchaseCount : <anonymous< td=""><td>> (complex) Repeating index: 1</td><td>Range in parsed input</td><td>: 259 - 261 Character Selection In In Row: 0 Column: 0</td><td>Byte Selection In Input Offset: 0 Length: 0</td></anonymous<>	> (complex) Repeating index: 1	Range in parsed input	: 259 - 261 Character Selection In In Row: 0 Column: 0	Byte Selection In Input Offset: 0 Length: 0
•				•

2. In the File Selection dialog, select the "Select an input file from the file system" option.

Click on the Browse button.

File Selection		
Select an input file:		
MessageModelli	ngLibrary	
) Seect an input file fi	rom the file system	Browse
?	OK	Cancel

3. Navigate to the "C:\student10\MessageModeling\data" directory and select the "purchases_MALFORMED.dat" file.

File Selection	
Select an input file:	
MessageModellingLibrary MessageModellingLibrary MessageModellingLibrary IBMdefined MessageModellingLibrary ImportFiles MessageModellingLibrary ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles ImportFiles Imp	
Select an input file from the file system C:\student10\MessageModeling\data\purchase	Browse
ОК	Cancel

Click OK.

4. In the "Offset" textbox, enter "298", and scroll right to find the highlighted character (byte 298 will be highlighted in blue).

Note that at this position (the "Price" element position) there is a string "MALFORMED!" instead of the expected decimal number.

Now click on the "Run parser" button to test the message model (green arrow as highlighted below).

😪 Navigator 🖹 Problems 🖺 DFDL Test - Parse 🛛 🗟 DFDL Test - Serialize 🚯 DFDL Test - Trace		Pi 🖬 🗈 🖳 🤣 📗 🗵 🔍 🗖
DFDL Test - Parse: Runs the DFDL parser with the provided physical input data and selected message, and updates the logical instance view v		sult of the parse.
Status: Ready		
_ Input		
Data: C:\student10\MessageModeling\data\purchases_MALI Browse Encoding (code page): UTF-8	• M	lessage: PURCHASES (/MessageModellingLibrary/PURCHA
□ Parsed Input		
Characters		
1 04082008 00001Beer 06WALFORMED!00002Chips		01000000225
	T	Put-of-month
Selected: PurchaseCount : < Anonymous > (complex) Repeating index: 1 Range in parsed input: -11 Row: 1 Column: 2	Input	Offset: 298 enoth: 0
Selected, Parchasedount, CAnonymous / (complex) Repeating index. 1 Range in parsed input, 11-1 Row, 1 Column, 2		chart 250 rengen 0
4		•

5. An error message will appear with the cause of the failed parsing.

DFDL 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. Processing Errors CTDP3053E: Zoned text to number conversion error for element "Price": MALFORMEDI PracedbataRegion[SimpleContent, startOffset = 298, length = 10, scd = #xscd/type::PURCHASES/model::sequence/schemaElement::Purchase/type::0/model::sequence/schemaElement::Price)] errors received during parsing are hiphighted in the parsed input section of the DFDL Trace View toolbar button, or click here. To view the parala 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

Inspect the "DFDL Test - Logical Instance", you will see that the parsed tree is not complete. 6.

Go to the Purchase element, expand it, and check that it was correctly parsed until the "Amount" element. The following field "Price" is empty.

Tree View XML View						
Name	Туре	Value				
PURCHASES						
REQUEST_TYPE	xs:string	Α				
RET_CODE	xs:string	00				
CustomerId	xs:string	12345678				
CustomerLastName	xs:string	Griffin				
CustomerFirstName	xs:string	Peter				
CustomerCompany	xs:string	Pawtucket Brew				
CustomerAddr1	xs:string	31 Spooner st.				
CustomerAddr2	xs:string	456 1st av.				
CustomerCity	xs:string	Quahog				
CustomerState	xs:string	Rhode Island				
CustomerCountry	xs:string	USA				
CustomerMailCode	xs:string	12312				
CustomerPhone	xs:string	123-123-1234				
CustomerLastUpdateD	xs:string	04082008				
PurchaseCount	xs:unsignedShort	8196				
 Purchase 						
PurchaseId	xs:unsignedInt	1				
ProductName	xs:string	Beer				
Amount	xs:unsignedShort	6				

Now you will use the "DFDL Test - Trace" view, to better understand what the problem was. 7. Click on the "DFDL Test - Trace" view.

· · · · · · · · · · · · · · · · · · ·			
😪 Navigator 🔝 Problems 🖹 DFDL Test - Parse 💥 🗟 DFDL Test - Serialize	DFDL Test - Trace	0	🖣 🖬 🕼 🤐 🤣 🔢 🛛 🖓 🗖
DFDL Test - Parse: Runs the DFDL parser with the provided physical input data an	d selected message, and	updates the logical instance view with the re	sult of the parse.
Status: 😡 Parsing completed with processing errors: Wed Mar 11 12:14:40 GMT	2015		
_ Input			
Data: C:\student10\MessageModeling\data\purchases_MALFORMED.dat	Browse Encoding	(code page): UTF-8	Message: PURCHASES (/MessageModellin
Parsed Input			
Characters			
123-123-123-123-123-123-123-123-123-123-	-1234 N	4082008 00001Beer	

8. In the "DFDL Test - Trace" view, you will find an execution log of the parsing activities.

At the end of the trace, there are colored lines with the found error.

		_
🔁 Navigator 🖹 Problems 🖹	🖕 DFDL Test - Parse 🔓 DFDL Test - Serialize 🖺 DFDL Test - Trace 🛛	3
DFDL Trace Console		
11 Mar 2015 12:14:40	<pre>info: Offset: 298. Starting to process element 'Price'. [dfdl = /MessageModellingLibrary/PURCHASES.xsd, scd = #xscd(/type::PURCHASES/model::sequence/schemaElement::Purchase/ty</pre>	•
11 Mar 2015 12:14:40	info: Offset: 308. The simple content region of element 'Price' does not match the literal nil value. [dfdl = /MessageModellingLibrary/PURCHASES.xsd, scd = #xscd(/type::PURCHASES/model::sequence/schemaElement::Purchase/ty	
11 Mar 2015 12:14:40	<pre>info: Offset: 298. Found specified length value 'MALFORMED!' for element 'Price'. The length was 10 bytes [dfdl = /MessageModellingLibrary/PURCHASES.xsd, scd = #xscd(/type::PURCHASES/model::sequence/schemaElement::Purchase/ty</pre>	
11 Mar 2015 12:14:40	error: CTDP3053E: Zoned text to number conversion error for element 'Price': MALFORMED!	1
11 Mar 2015 12:14:40	fatal: CTDP3053E: Zoned text to number conversion error for element 'Price': MALFORMED!	
		•

9. Look at the lines before the error:

😪 Navigator 🖹 Problems 🕒	🔓 DFDL Test - Parse 🔄 DFDL Test - Serialize 🕼 DFDL Test - Trace 🕴	. 🔠 🛃 🗖 🗖
DFDL Trace Console		
11 Mar 2015 12:14:40	<pre>info: Offset: 298. Starting to process element 'Price'. [dfdl = /MessageModellingLibrary/PURCHASES.xsd, scd = #xscd(/type::PURCHASES/model::sequence/schemaElement</pre>	▲ t::Purchase/ty
11 Mar 2015 12:14:40	<pre>info: Offset: 308. The simple content region of element 'Price' does not match the literal nil value. [dfdl = /MessageModellingLibrary/PURCHASES.xsd, scd = #xscd(/type::PURCHASES/model::sequence/schemaElement</pre>	t::Purchase/ty
11 Mar 2015 12:14:40	<pre>info: Offset: 298. Found specified length value 'MALFORMED!' for element 'Price'. The length was 10 bytes [dfdl = /MessageModellingLibrary/PURCHASES.xsd, scd = #xscd(/type::PURCHASES/model::sequence/schemaElement</pre>	 t::Purchase/ty
11 Mar 2015 12:14:40	error: CTDP3053E: Zoned text to number conversion error for element 'Price': MALFORMED!	
11 Mar 2015 12:14:40	fatal: CTDP3053E: Zoned text to number conversion error for element 'Price': MALFORMED!	
		-

The first line states that it is starting to process the Price element.

In the third, it has found a string "MALFORMED!" as the value of the element.

Then the parser tries to convert the string to a decimal number, and an error appears.

10. Back in the DFDL Editor, scroll to the "Price" element.

Note that it has an error icon next to its name.

0 PUF	RCHASES.xs	id 🖾 🗹 PURCHASE	ES.cpy				
Test P	E Model	Test Serialize Model	Show pr	perties	A show all sections	», •	
	arse moder	e ProductName	Show pro	<picx_< td=""><td>_string></td><td>-</td><td>1</td></picx_<>	_string>	-	1
1		e Amount		<pic9-< td=""><td>Display-Zonedsho</td><td>ort></td><td>1</td></pic9-<>	Display-Zonedsho	ort>	1
6		e Price		<pic9-< td=""><td>Display-Zoneddec</td><td>cimal ></td><td>1</td></pic9-<>	Display-Zoneddec	cimal >	1
1	e RETURN_COMMENT <picx_string></picx_string>					1	
Add a Local Element							

11. Place the cursor on the error icon and a message will appear, showing the same error cause you saw in the trace view.

📵 PURCHASES.xsd 🖾 🗹 PURCHASES.cpy	
Test Parse Model Test Serialize Model Show p	roperties Show all sections 3
e ProductName	<picx_string> 1 <pic9-display-zoned_short> 1</pic9-display-zoned_short></picx_string>
😡 🛛 e Price	<pic9-display-zoned_decimal></pic9-display-zoned_decimal>
Image: string comparison of the string in the string is a string in the string comparison of the string is a string in the string in the string is a string in the string in th	

END OF LAB GUIDE