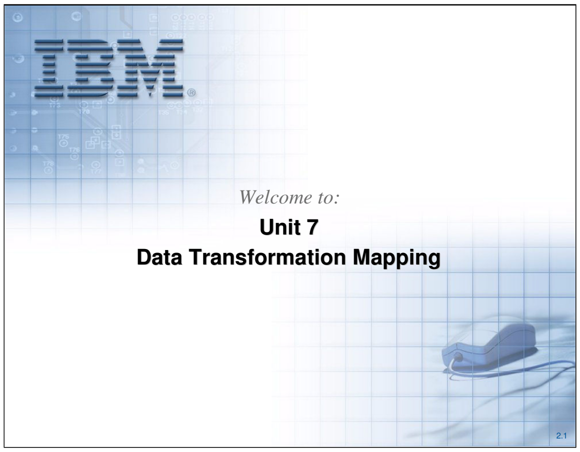


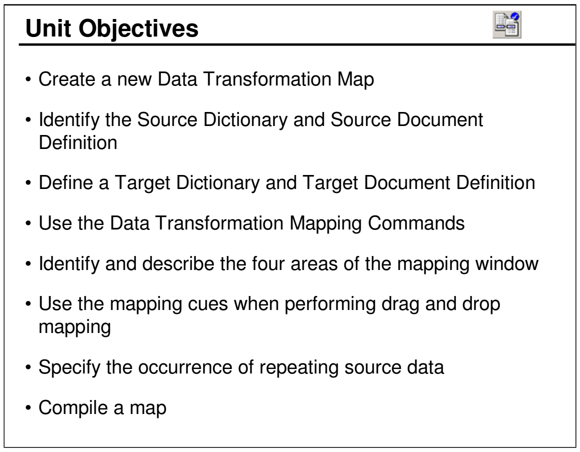
Welcome to:

Unit 7

Data Transformation Mapping

2.1

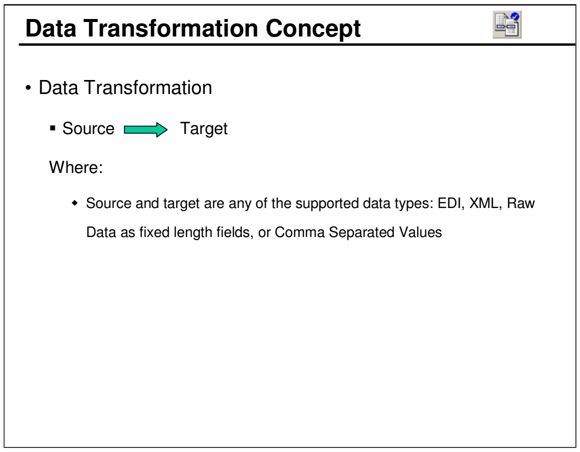
This unit addresses basic Data Transformation mapping.



Unit Objectives

- Create a new Data Transformation Map
- Identify the Source Dictionary and Source Document Definition
- Define a Target Dictionary and Target Document Definition
- Use the Data Transformation Mapping Commands
- Identify and describe the four areas of the mapping window
- Use the mapping cues when performing drag and drop mapping
- Specify the occurrence of repeating source data
- Compile a map

Creating a map involves naming it, identifying the input and output formats and defining map characteristics. Most mapping will be performed via the “drag and drop” technique, but there are many conditions under which commands are entered to provide conditional mapping and control over the translation process.

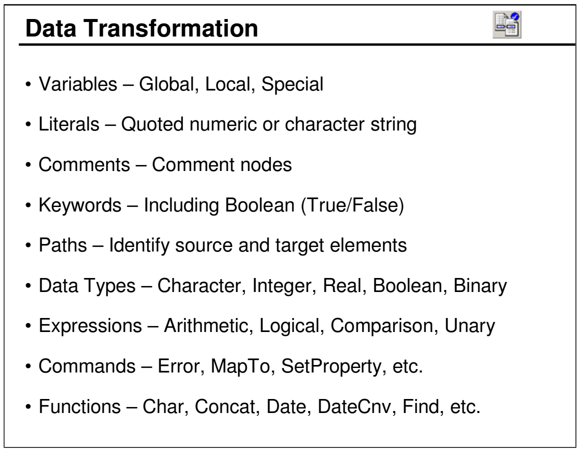


Data Transformation Concept

- Data Transformation
  - Source → Target

  Where:
    - Source and target are any of the supported data types: EDI, XML, Raw Data as fixed length fields, or Comma Separated Values

Data Transformation mapping is based on translation of source data (input) to target data (output) where the source and target data may be defined as *Record Oriented Data, XML,* or *EDI*.

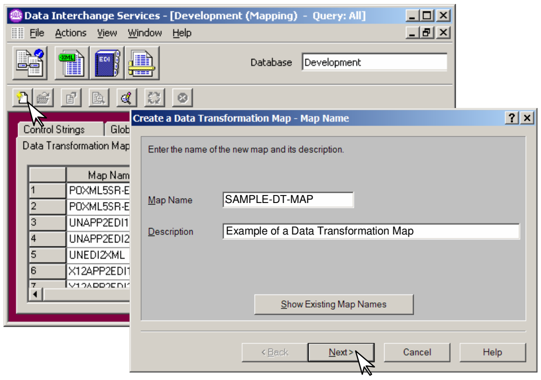


Data Transformation

- Variables – Global, Local, Special
- Literals – Quoted numeric or character string
- Comments – Comment nodes
- Keywords – Including Boolean (True/False)
- Paths – Identify source and target elements
- Data Types – Character, Integer, Real, Boolean, Binary
- Expressions – Arithmetic, Logical, Comparison, Unary
- Commands – Error, MapTo, SetProperty, etc.
- Functions – Char, Concat, Date, DateCnv, Find, etc.

Data Transformation maps include extensive programming commands and functions. All commands and functions will be addressed in this unit.

Creating a New Data Transformation Map



Data Interchange Services - [Development (Mapping) - Query: All]

Database: Development

Create a Data Transformation Map - Map Name

Enter the name of the new map and its description.

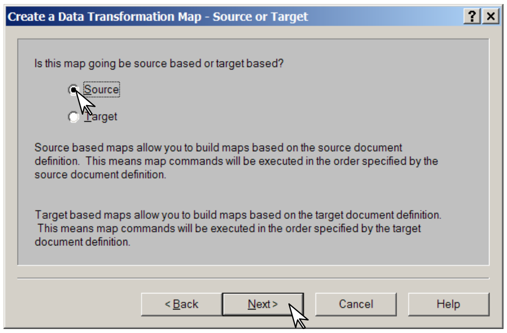
Map Name: SAMPLE-DT-MAP

Description: Example of a Data Transformation Map

	Map Nam
1	POXML5SR-E
2	POXML5SR-E
3	UNAPP2EDI1
4	UNAPP2EDI2
5	UNEDI2XML
6	X12APP2EDI1

Note that the map is being produced in the database selected. The new icon may be selected or you can select File/New to create a new map. The map name may be up to 16 characters long. The description may only be 50 characters long, but additional information may be stored under the Comments tab.

Source or Target Based



Create a Data Transformation Map - Source or Target

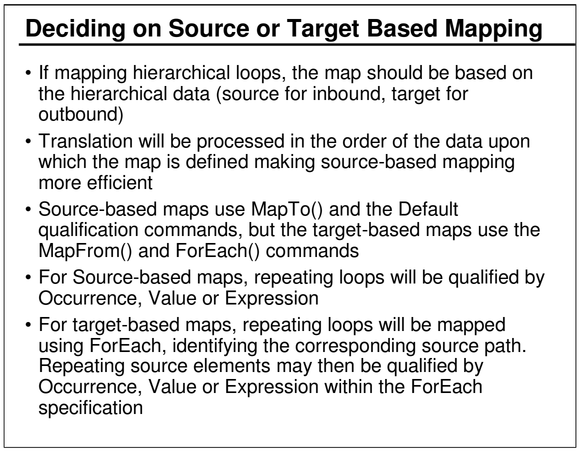
Is this map going be source based or target based?

● Source
○ Target

Source based maps allow you to build maps based on the source document definition. This means map commands will be executed in the order specified by the source document definition.

Target based maps allow you to build maps based on the target document definition. This means map commands will be executed in the order specified by the target document definition.

Data Transformation maps allow commands to be imbedded within either the source or target document definition.

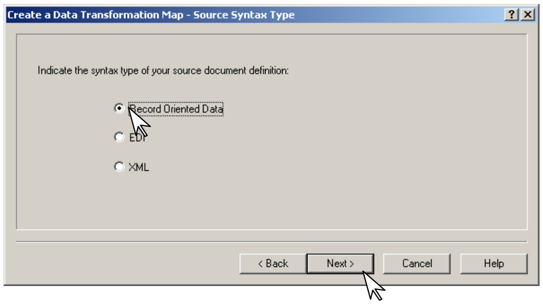


Deciding on Source or Target Based Mapping

- If mapping hierarchical loops, the map should be based on the hierarchical data (source for inbound, target for outbound)
- Translation will be processed in the order of the data upon which the map is defined making source-based mapping more efficient
- Source-based maps use MapTo() and the Default qualification commands, but the target-based maps use the MapFrom() and ForEach() commands
- For Source-based maps, repeating loops will be qualified by Occurrence, Value or Expression
- For target-based maps, repeating loops will be mapped using ForEach, identifying the corresponding source path. Repeating source elements may then be qualified by Occurrence, Value or Expression within the ForEach specification

It may take some experience with mapping to identify the better technique for each map.

Source Document Definition



Create a Data Transformation Map - Source Syntax Type

Indicate the syntax type of your source document definition:

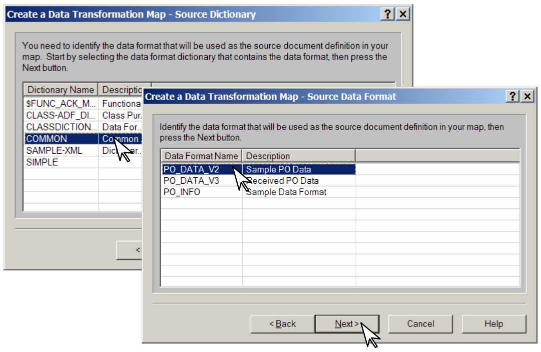
● Record Oriented Data

○ EDI

○ XML

The three radio buttons represent the three possible syntax types for the source document. XML is defined by a DTD Document Definition or Schema Document Definition".

Source Dictionary and Document


Create a Data Transformation Map - Source Dictionary
?
✕

You need to identify the data format that will be used as the source document definition in your map. Start by selecting the data format dictionary that contains the data format, then press the Next button.

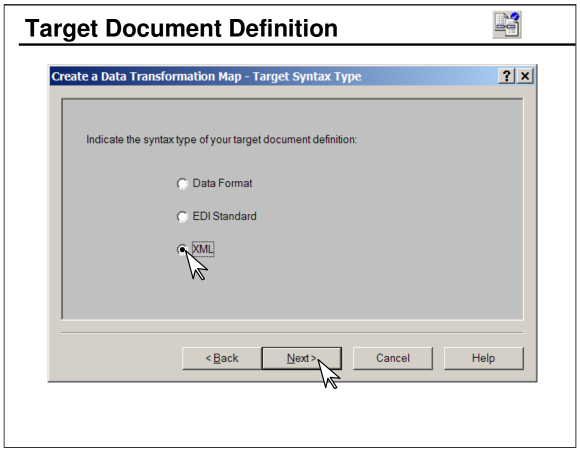
Dictionary Name	Descriptio
$FUNC_ACK_M...	Functiona
CLASS-ADF_DI...	Class Pur.
CLASSDICTION...	Data For..
COMMON	Common
SAMPLE-XML	Dictionar..
SIMPLE	

Create a Data Transformation Map - Source Data Format
?
✕

Identify the data format that will be used as the source document definition in your map, then press the Next button.

Data Format Name	Description
PO_DATA_V2	Sample PO Data
PO_DATA_V3	Received PO Data
PO_INFO	Sample Data Format

All document definitions are contained within dictionaries. The DIS Client will display all the dictionaries for the source syntax type selected. Once a dictionary is selected, the Client will display a list of all document definitions within that dictionary. You will identify both the dictionary and document definition for both the source and target document.



Target Document Definition

Create a Data Transformation Map - Target Syntax Type

Indicate the syntax type of your target document definition:

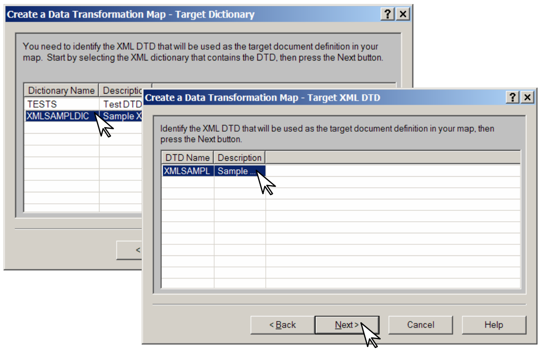
○ Data Format

○ EDI Standard

● XML

Here you again select the syntax type. You have the same three choices for the target as the source. The target and source documents may be of the same type.

Target Dictionary and Document



Create a Data Transformation Map - Target Dictionary

You need to identify the XML DTD that will be used as the target document definition in your map. Start by selecting the XML dictionary that contains the DTD, then press the Next button.

Dictionary Name	Descriptic
TESTS	Test DTD
XMLSAMPLDIC	Sample X

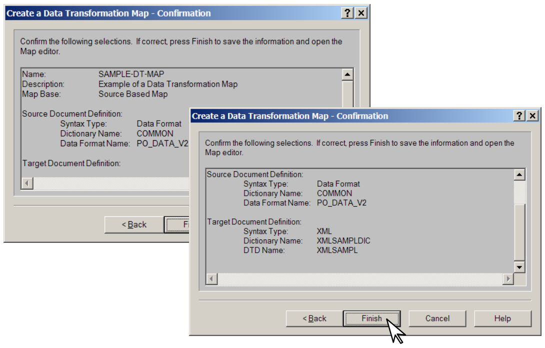
Create a Data Transformation Map - Target XML DTD

Identify the XML DTD that will be used as the target document definition in your map, then press the Next button.

DTD Name	Description
XMLSAMPL	Sample ...

The DIS Client will display all the dictionaries for the target syntax type selected. Once a dictionary is selected, the Client will display a list of all document definitions within that dictionary.

Confirmation



Create a Data Transformation Map - Confirmation

Confirm the following selections. If correct, press Finish to save the information and open the Map editor.

```
Name:               SAMPLE-DT-MAP
Description:        Example of a Data Transformation Map
Map Base:           Source Based Map

Source Document Definition:
          Syntax Type:        Data Format
          Dictionary Name:    COMMON
          Data Format Name:   PO_DATA_V2

Target Document Definition:
```

Create a Data Transformation Map - Confirmation

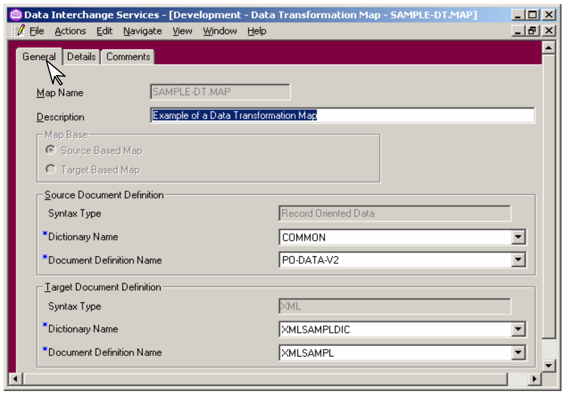
Confirm the following selections. If correct, press Finish to save the information and open the Map editor.

```
Source Document Definition:
          Syntax Type:        Data Format
          Dictionary Name:    COMMON
          Data Format Name:   PO_DATA_V2

Target Document Definition:
          Syntax Type:        XML
          Dictionary Name:    XMLSAMPLDIC
          DTD Name:           XMLSAMPL
```

At this point you can click “Back” to change any of these selections. Once “Finish” is clicked, you will not be able to change the source or target syntax types, but will be able to change the document dictionary and definition selected.

The General Tab


Data Interchange Services - [Development - Data Transformation Map - SAMPLE-DT.MAP]
File Actions Edit Navigate View Window Help
General
Details
Comments

Map Name: SAMPLE-DT.MAP

Description: Example of a Data Transformation Map

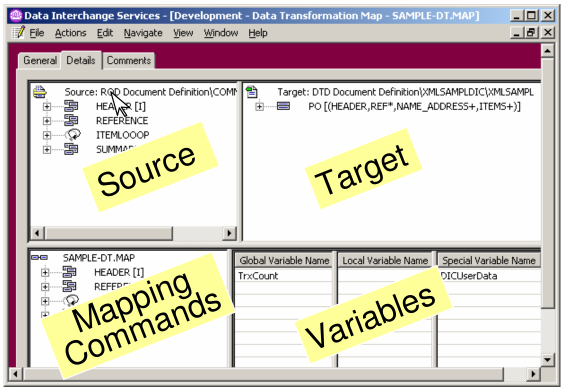
Map Base
● Source Based Map
○ Target Based Map

Source Document Definition
Syntax Type: Record Oriented Data
*Dictionary Name: COMMON
*Document Definition Name: PO-DATA-V2

Target Document Definition
Syntax Type: XML
*Dictionary Name: XMLSAMPLDIC
*Document Definition Name: XMLSAMPL

The unavailable areas cannot be changed on this tag page. Also, your map has already been saved at this point, so you don't have to map now.

Data Transformation Mapping Details



Data Interchange Services - [Development - Data Transformation Map - SAMPLE-DT.MAP]
File Actions Edit Navigate View Window Help
General Details Comments

Source: RCD Document Definition\COMM
HEADER [I]
REFERENCE
ITEMLOOOP
SUMMAR

Target: DTD Document Definition\XMLSAMPLDIC\XMLSAMPL
PO [(HEADER,REF*,NAME_ADDRESS+,ITEMS+)]

SAMPLE-DT.MAP
HEADER [I]
REFER

Global Variable Name	Local Variable Name	Special Variable Name
TrxCount		DICUserData

Source

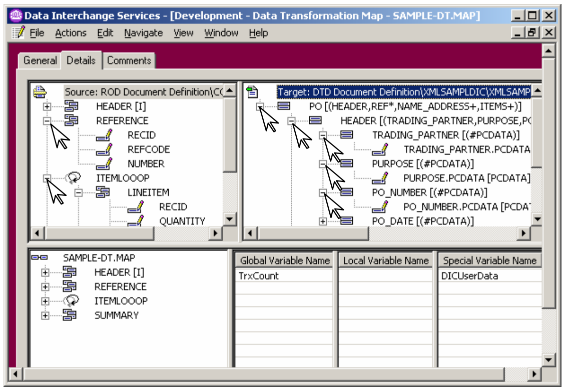
Target

Mapping Commands

Variables

Note that the four panes in this window may be resized. While mapping you may find it convenient to resize regularly. At times it may be useful to have the command pane occupy most of the screen.

Records and XML Elements Expanded


Data Interchange Services - [Development - Data Transformation Map - SAMPLE-DT.MAP]
File
Actions
Edit
Navigate
View
Window
Help
General
Details
Comments

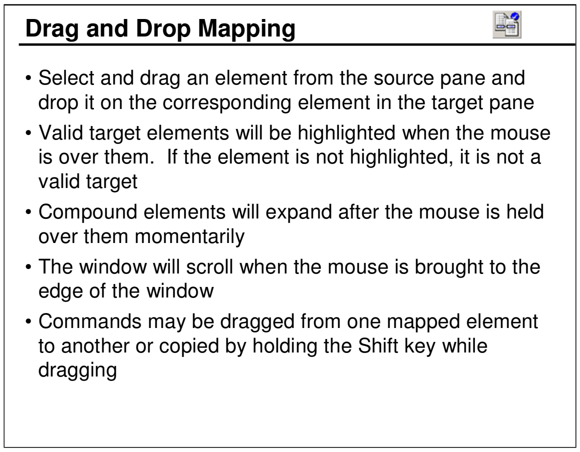
```
Source: ROD Document Definition\C(
  HEADER [I]
  REFERENCE
      RECID
      REFCODE
      NUMBER
  ITEMLOOOP
      LINEITEM
          RECID
          QUANTITY
```

```
Target: DTD Document Definition\XMLSAMPLDIC\XMLSAMP
  PO [(HEADER,REF*,NAME_ADDRESS+,ITEMS+)]
      HEADER [(TRADING_PARTNER,PURPOSE,P(
          TRADING_PARTNER [(#PCDATA)]
              TRADING_PARTNER.PCDATA
          PURPOSE [(#PCDATA)]
              PURPOSE.PCDATA [PCDATA]
          PO_NUMBER [(#PCDATA)]
              PO_NUMBER.PCDATA [PCDA
          PO_DATE [(#PCDATA)]
```

```
SAMPLE-DT.MAP
  HEADER [I]
  REFERENCE
  ITEMLOOOP
  SUMMARY
```

Global Variable Name	Local Variable Name	Special Variable Name
TrxCount		DICUserData

Right click “Expand All”, “Collapse All”, and “Find” functions are available in source, target and mapping command windows. Use expand to expand the areas in which you are mapping. You will also find it convenient to collapse items to be able to focus on the area that is currently being mapped.

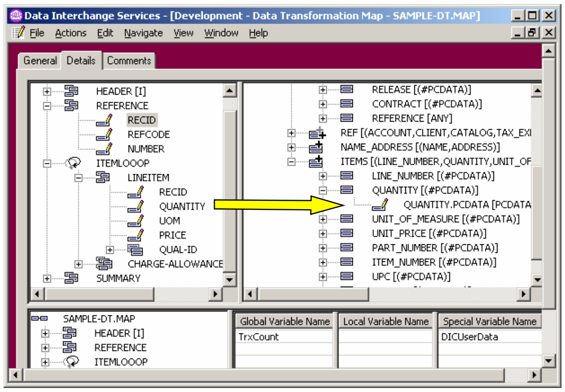


Drag and Drop Mapping

- Select and drag an element from the source pane and drop it on the corresponding element in the target pane
- Valid target elements will be highlighted when the mouse is over them. If the element is not highlighted, it is not a valid target
- Compound elements will expand after the mouse is held over them momentarily
- The window will scroll when the mouse is brought to the edge of the window
- Commands may be dragged from one mapped element to another or copied by holding the Shift key while dragging

The mouse pointer will turn to a “Not” symbol when the item being dragged is not over a valid area onto which it may be dropped.

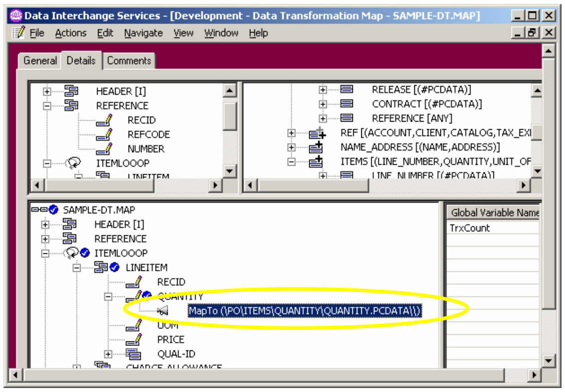
Drag and Drop Mapping



Global Variable Name	Local Variable Name	Special Variable Name
TrxCount		DICUserData

You can drag from source to target or target to source. You can also drag to the command pane. If variables are defined, they may also be dragged to the command pane.

The MapTo Command and Path


Data Interchange Services - [Development - Data Transformation Map - SAMPLE-DT.MAP]
File
Actions
Edit
Navigate
View
Window
Help
General
Details
Comments
Global Variable Name

```
HEADER [I]
REFERENCE
    RECID
    REFCODE
    NUMBER
ITEMLOOOP
    LINEITEM
```

```
        RELEASE [(#PCDATA)]
        CONTRACT [(#PCDATA)]
        REFERENCE [ANY]
    REF [(ACCOUNT,CLIENT,CATALOG,TAX_EXI
    NAME_ADDRESS [(NAME,ADDRESS)]
    ITEMS [(LINE_NUMBER,QUANTITY,UNIT_OF
        LINE_NUMBER [(#PCDATA)]
```

```
SAMPLE-DT.MAP
    HEADER [I]
    REFERENCE
    ITEMLOOOP
        LINEITEM
            RECID
            QUANTITY
                MapTo (\PO\ITEMS\QUANTITY\QUANTITY.PCDATA\\)
            UOM
            PRICE
            QUAL-ID
        CHARGE-ALLOWANCE
```

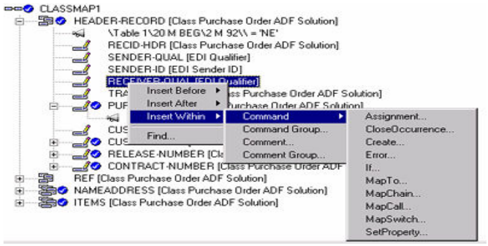
Global Variable Name
TrxCount

When drag and drop mapping is performed for a source-based map, a MapTo command is automatically generated. For a target-based map a MapFrom command is produced.

Mapping ICONS

Icon	Description
	Represents most mapping commands.
	Represents a command group.
	Represents conditional commands such as if()/ElseIf()/Else/EndIf.
	Represents the Qualify() and Default commands.
	Represents the ForEach() command.
	Represents the hierarchical loop mapping command *HLLevel()*.
	Represents the hierarchical loop mapping command *HLDefault*.
!	Positioned to the right of a mapping command icon to indicate there is an error in the mapping command. A description of the error will be contained within parenthesis following the mapping command.

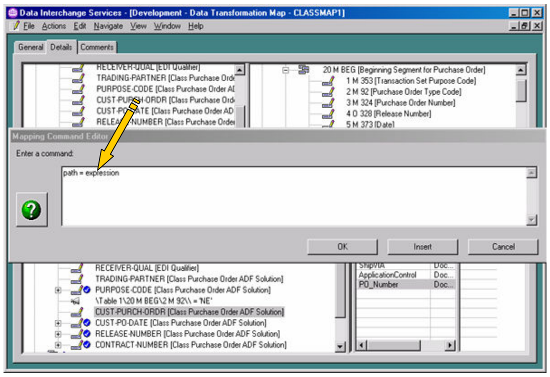
Entering Commands


CLASSMAP1
HEADER-RECORD [Class Purchase Order ADF Solution]
\Table 1\20 M BEG\2 M 92\\ = 'NE'
RECID-HDR [Class Purchase Order ADF Solution]
SENDER-QUAL [EDI Qualifier]
SENDER-ID [EDI Sender ID]
RECEIVER-QUAL [EDI Qualifier]
TRA
ss Purchase Order ADF Solution]
PUF
urchase Order ADF Solution]
CUS
CUS
RELEASE-NUMBER [Cla
CONTRACT-NUMBER [Class Purchase Order ADF
REF [Class Purchase Order ADF Solution]
NAMEADDRESS [Class Purchase Order ADF Solution]
ITEMS [Class Purchase Order ADF Solution]
Insert Before
Insert After
Insert Within
Find...
Command
Command Group...
Comment...
Comment Group...
Assignment...
CloseOccurrence...
Create...
Error...
If...
MapTo...
MapChain...
MapCall...
MapSwitch...
SetProperty...

- By right clicking in the mapping pane, commands may be inserted before, after, or within a field or element being mapped.
- In a complex map, you may also discover that the *Find* option is very useful.

While the “Drag and Drop” process causes MapTo and MapFrom commands to be generated automatically, commands may also be entered at any point in the command pane.

Dragging Elements to the Command Edit



HECEIVER-QUAL [EDI Qualifier]
TRADING-PARTNER [Class Purchase Ord
PURPOSE-CODE [Class Purchase Order AI
CUST-PURCH-ORDR [Class Purchase Ord
CUST-PO-DATE [Class Purchase Order AD
RELEASE-NUMBER [Class Purchase Order

20 M BEG [Beginning Segment for Purchase Order]
1 M 353 [Transaction Set Purpose Code]
2 M 92 [Purchase Order Type Code]
3 M 324 [Purchase Order Number]
4 O 328 [Release Number]
5 M 373 [Date]

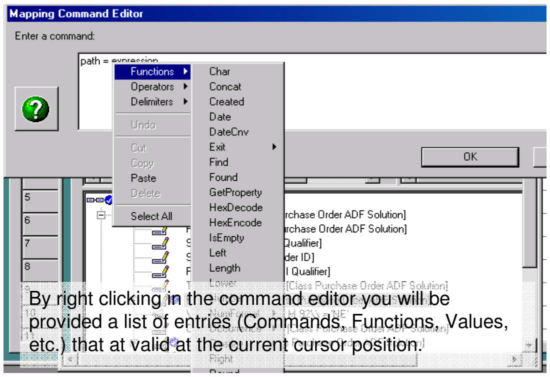
```
path = expression
```

RECEIVER-QUAL [EDI Qualifier]
TRADING-PARTNER [Class Purchase Order ADF Solution]
PURPOSE-CODE [Class Purchase Order ADF Solution]
\Table 1\20 M BEG\2 M 92\\ = 'NE'
CUST-PURCH-ORDR [Class Purchase Order ADF Solution]
CUST-PO-DATE [Class Purchase Order ADF Solution]
RELEASE-NUMBER [Class Purchase Order ADF Solution]
CONTRACT-NUMBER [Class Purchase Order ADF Solution]

ShipVIA	Doc...
ApplicationControl	Doc...
PO_Number	Doc...

You can drag to the command editor from the source pane, the target pane, the variable pane, or the command pane.

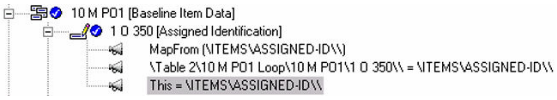
Command Editor Selections



By right clicking in the command editor you will be provided a list of entries (Commands, Functions, Values, etc.) that at valid at the current cursor position.

If you right click on an argument to a function, the Client will provide you with a list of possible values for that argument. In this example, when “expression” is right-clicked, these menus are displayed.

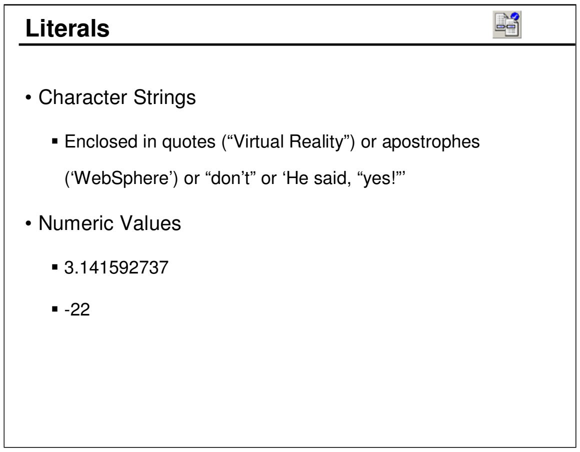
Mapping Options



```
10 M P01 [Baseline Item Data]
    1 O 350 [Assigned Identification]
        MapFrom (\ITEMS\ASSIGNED-ID\\)
        \Table 2\10 M P01 Loop\10 M P01\1 O 350\\ = \ITEMS\ASSIGNED-ID\\
        This = \ITEMS\ASSIGNED-ID\\
```

- Any of the above provide the same mapping.
- *This* refers to the element within which the command is located.
- *This* is used as a target for target-based maps,
  or a source for source-based maps.

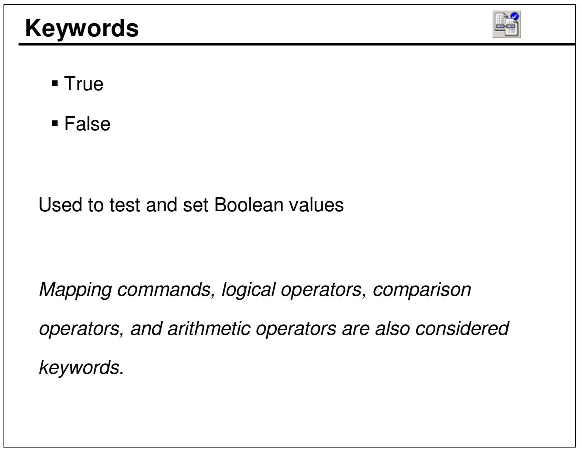
There are many ways to create a map and several methods may be available for any mapping. Be aware of mapping methods and find those that work best for you.



Literals

- Character Strings
  - Enclosed in quotes (“Virtual Reality”) or apostrophes (‘WebSphere’) or “don’t” or ‘He said, “yes!”’
- Numeric Values
  - 3.141592737
  - -22

Literals may be used to set values for output elements or fields or may be used to compare element or field values in logical expressions.



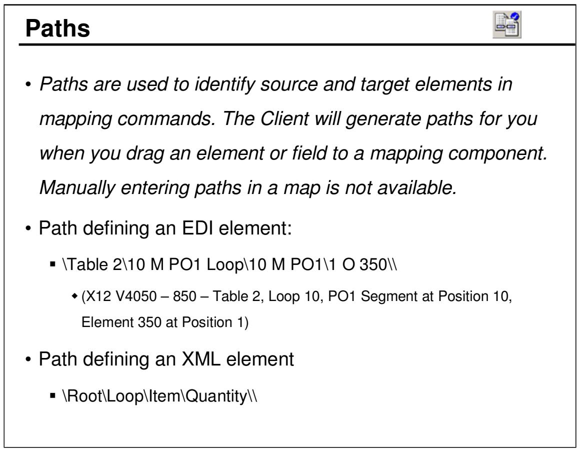
Keywords

- True
- False

Used to test and set Boolean values

Mapping commands, logical operators, comparison operators, and arithmetic operators are also considered keywords.

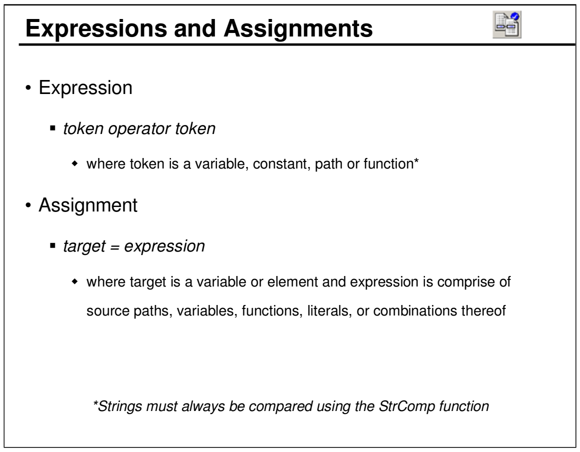
Keywords are not coded within quotes.



Paths

- *Paths are used to identify source and target elements in mapping commands. The Client will generate paths for you when you drag an element or field to a mapping component. Manually entering paths in a map is not available.*
- Path defining an EDI element:
  - \Table 2\10 M PO1 Loop\10 M PO1\1 O 350\\
    - (X12 V4050 – 850 – Table 2, Loop 10, PO1 Segment at Position 10, Element 350 at Position 1)
- Path defining an XML element
  - \Root\Loop\Item\Quantity\\

Slashes identify each new identifier in the path. The Path Definition is terminated with two slashes.



Expressions and Assignments

- Expression
  - *token operator token*
    - where token is a variable, constant, path or function*
- Assignment
  - *target = expression*
    - where target is a variable or element and expression is comprise of source paths, variables, functions, literals, or combinations thereof

**Strings must always be compared using the StrComp function*

Expressions are not “stand-alone” but are components of commands. Examples of expressions include:

X+Y-Z*2*(A/B)

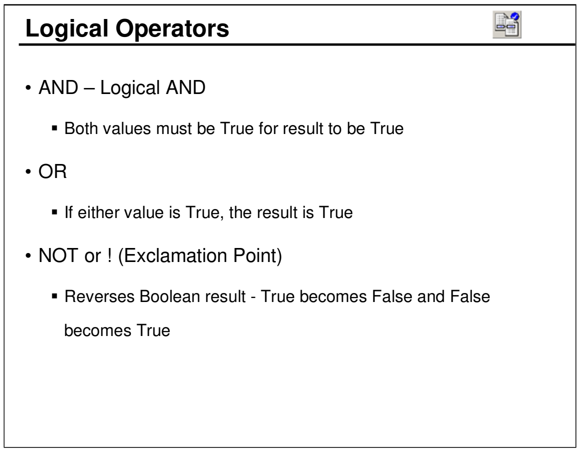
TOTAL = QUANTITY (*This is a logical expression, evaluated as true or false.)*

ROUND(VALUE) (*This expression references a function.)*

Note that, in addition to variable references, paths to source or target elements may be used at any point in an expression.

Assignment statements may take the form:

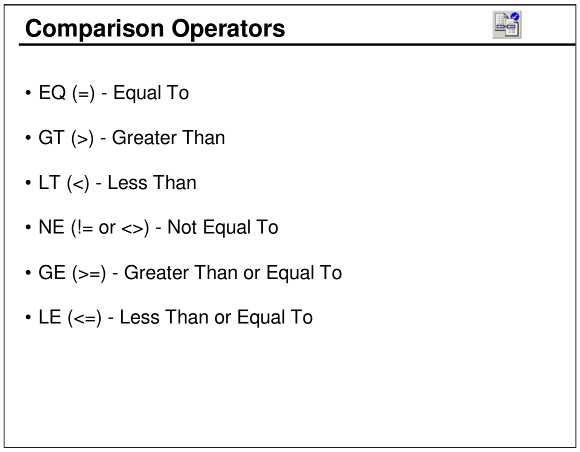
target-path = source-path



Logical Operators

- AND – Logical AND
  - Both values must be True for result to be True
- OR
  - If either value is True, the result is True
- NOT or ! (Exclamation Point)
  - Reverses Boolean result - True becomes False and False becomes True

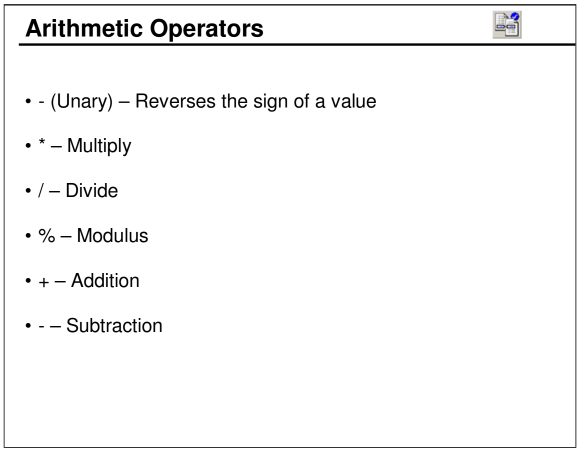
Logical expressions always result in a True or False value.



Comparison Operators

- EQ (=) - Equal To
- GT (>) - Greater Than
- LT (<) - Less Than
- NE (!= or <>) - Not Equal To
- GE (>=) - Greater Than or Equal To
- LE (<=) - Less Than or Equal To

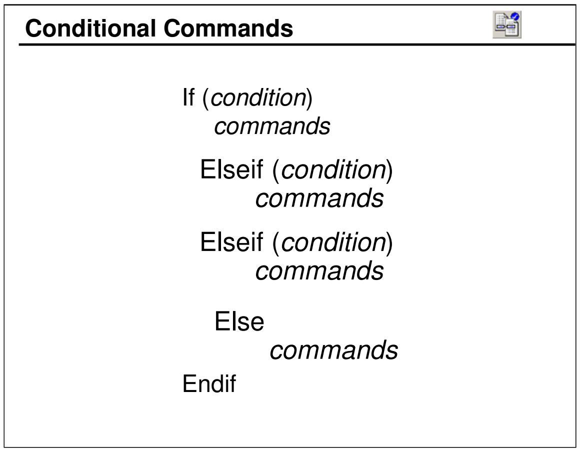
Either symbols or the two-character codes may be used. They may also be mixed in a single expression.



Arithmetic Operators

- - (Unary) – Reverses the sign of a value
- * – Multiply
- / – Divide
- % – Modulus
- + – Addition
- - – Subtraction

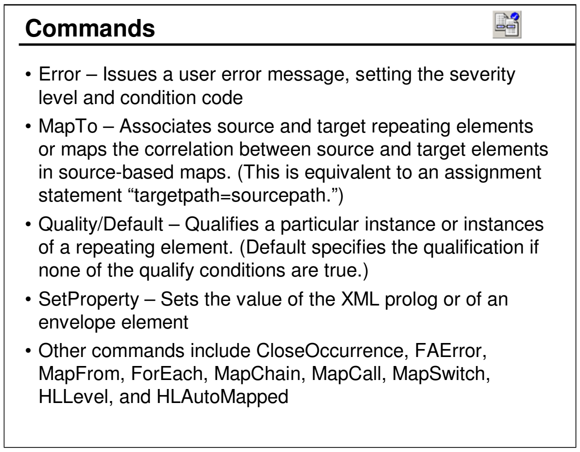
Modulus is the “remainder” after dividing operand 1 by operand 2.



Conditional Commands

```
If (condition)
   commands
  Elseif (condition)
       commands
  Elseif (condition)
       commands
   Else
        commands
Endif
```

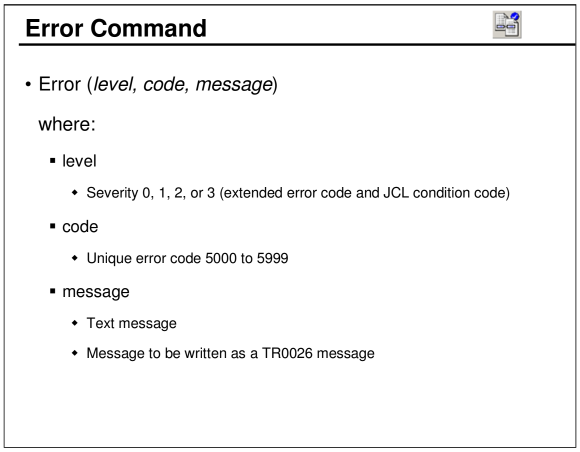
The *EndIf* is created automatically. The commands to be executed when the logical condition is true are inserted within the *If* command. The *ElseIf* and *Else* commands are inserted after the *If* command. These keywords are not case sensitive.



Commands

- Error – Issues a user error message, setting the severity level and condition code
- MapTo – Associates source and target repeating elements or maps the correlation between source and target elements in source-based maps. (This is equivalent to an assignment statement “targetpath=sourcepath.”)
- Quality/Default – Qualifies a particular instance or instances of a repeating element. (Default specifies the qualification if none of the qualify conditions are true.)
- SetProperty – Sets the value of the XML prolog or of an envelope element
- Other commands include CloseOccurrence, FAError, MapFrom, ForEach, MapChain, MapCall, MapSwitch, HLLevel, and HLAutoMapped

These are a few sample commands. This unit will address all the commands and show sample use within a map. Right-clicking in the map will provide a drop-down list of commands and functions appropriate to the current syntax.



Error Command

- Error (*level, code, message*)

  where:

  - level
    - Severity 0, 1, 2, or 3 (extended error code and JCL condition code)
  - code
    - Unique error code 5000 to 5999
  - message
    - Text message
    - Message to be written as a TR0026 message

The severity code definitions are:

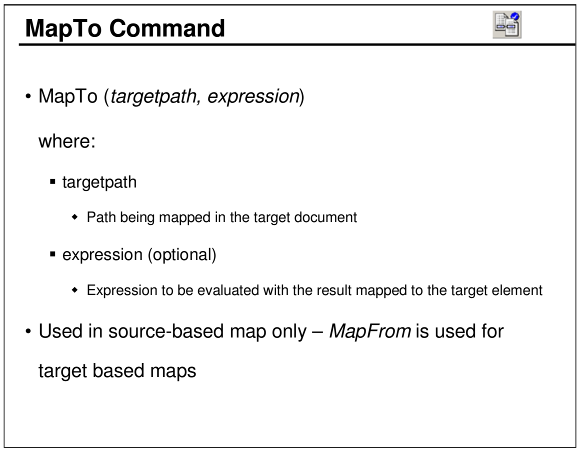
0 – no error

1 – simple element error (field or element)

2 – compound element error (segment, record, loop, etc.)

3 – transaction level error

Remember to place the text message in quotes if it is a literal. It may also be an expression or variable, however.



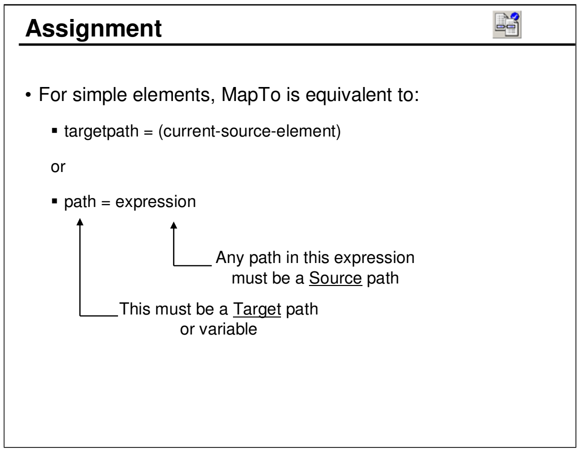
MapTo Command

- MapTo (*targetpath, expression*)

  where:

  - targetpath
    - Path being mapped in the target document
  - expression (optional)
    - Expression to be evaluated with the result mapped to the target element
- Used in source-based map only – *MapFrom* is used for target based maps

The *MapTo* command (without *expression)* is produced automatically when drag-and-drop mapping is used for a source-based map. The *MapTo* command may also be manually entered or selected from a drop-down list. It is the equivalent of the assignment statement: *target-path* = *expression*. If the assignment command is within an element mapping, *MapTo* is the equivalent of *This* = *expression.*



Assignment

- For simple elements, MapTo is equivalent to:
  - targetpath = (current-source-element)

  or

  - path = expression

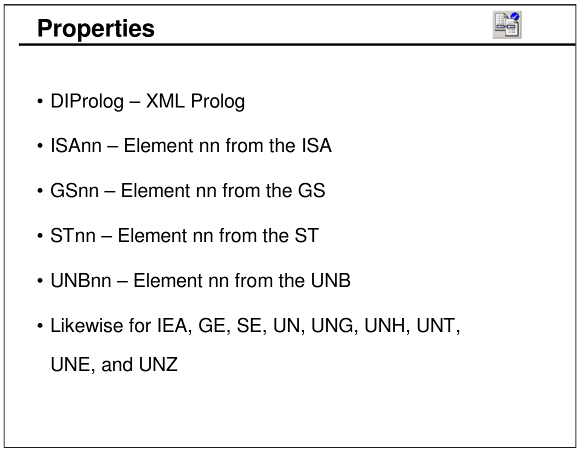
Any path in this expression must be a Source path

This must be a Target path or variable

Typically an assignment statement is produced by right-clicking on the element in the command pane, then selecting *Insert within / Command / Assignment.*

The target is then dragged over the word *path* and *expression* is replaced by typing an expression or dragging an source element or variable.

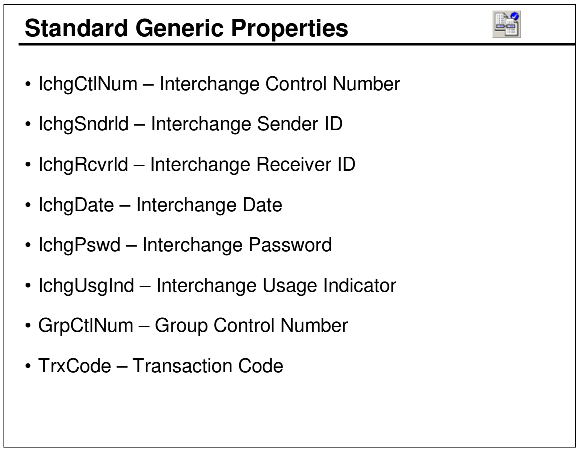
Hint: A quick way to produce an assignment statement is to drag any variable to the command pane element being mapped. You can then replace the variable name with the expression or path defining the source data for the mapping.



Properties

- DIProlog – XML Prolog
- ISAnn – Element nn from the ISA
- GSnn – Element nn from the GS
- STnn – Element nn from the ST
- UNBnn – Element nn from the UNB
- Likewise for IEA, GE, SE, UN, UNG, UNH, UNT, UNE, and UNZ

The use of properties will be discussed.

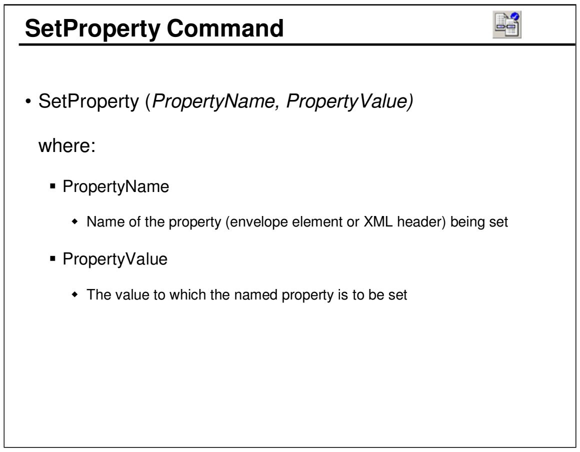


Standard Generic Properties

- IchgCtlNum – Interchange Control Number
- IchgSndrId – Interchange Sender ID
- IchgRcvrId – Interchange Receiver ID
- IchgDate – Interchange Date
- IchgPswd – Interchange Password
- IchgUsgInd – Interchange Usage Indicator
- GrpCtlNum – Group Control Number
- TrxCode – Transaction Code

A complete list of Standard Generic Properties may be found in the Appendix B of the Client User's Guide.

IchgSndrId is an alternative to ISA06; IchgRcvrId is an alternative to ISA08, etc.



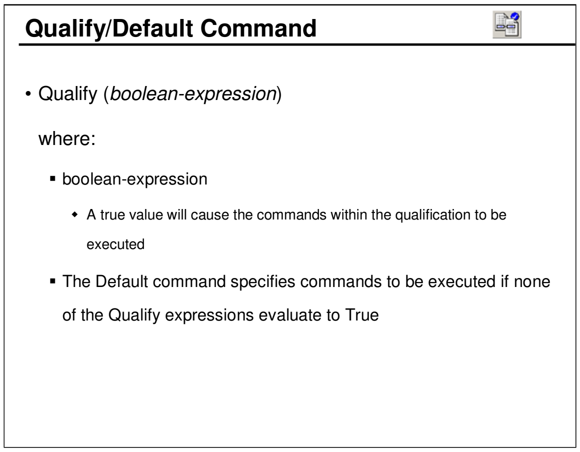
SetProperty Command

- SetProperty (*PropertyName, PropertyValue)*

  where:

  - PropertyName
    - Name of the property (envelope element or XML header) being set
  - PropertyValue
    - The value to which the named property is to be set

SetProperty may be used to assign a value to an envelope element when EDI standard data is being produced. It may be used to set the XML prolog when XML data is being produced.



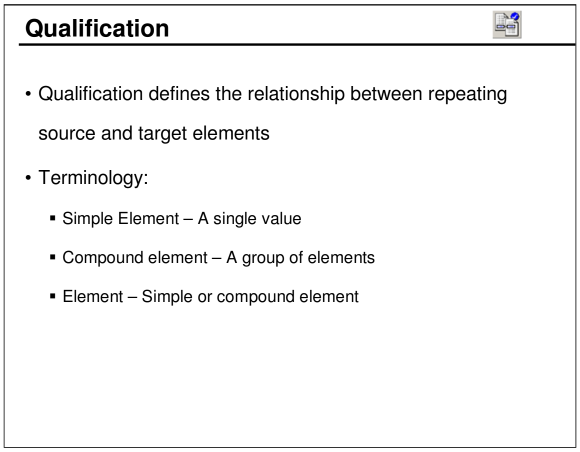
Qualify/Default Command

- Qualify (*boolean-expression*)

  where:

  - boolean-expression
    - A true value will cause the commands within the qualification to be executed
  - The Default command specifies commands to be executed if none of the Qualify expressions evaluate to True

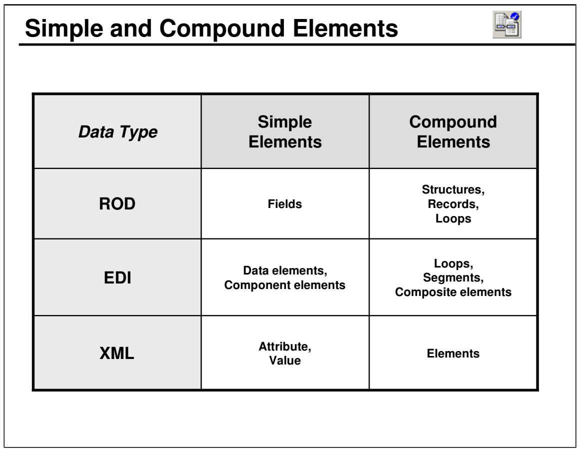
To insert a new qualify after the current one, right click on the existing *Qualify* and select *Qualify*. If you select *Qualify* at the compound element level, the new qualification will be inserted <u>before</u> the first qualification for that compound element. The *Default* command is used to provide mappings for the condition when none of the *Qualify* commands are selected based on the qualification criteria.



Qualification

- Qualification defines the relationship between repeating source and target elements
- Terminology:
  - Simple Element – A single value
  - Compound element – A group of elements
  - Element – Simple or compound element

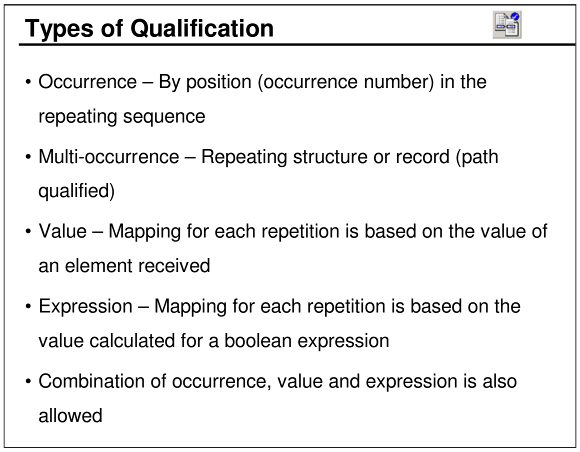
Because different data types use different terminology, standard convention for reference within the Client is used.



Simple and Compound Elements

Data Type	Simple Elements	Compound Elements
ROD	Fields	Structures, Records, Loops
EDI	Data elements, Component elements	Loops, Segments, Composite elements
XML	Attribute, Value	Elements

Compound elements are elements that may contain simple elements.

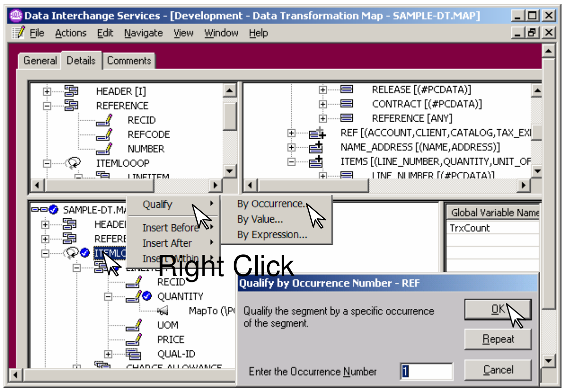


Types of Qualification

- Occurrence – By position (occurrence number) in the repeating sequence
- Multi-occurrence – Repeating structure or record (path qualified)
- Value – Mapping for each repetition is based on the value of an element received
- Expression – Mapping for each repetition is based on the value calculated for a boolean expression
- Combination of occurrence, value and expression is also allowed

When data repeats, you must specify in the map how to handle each occurrence of the repeating data.

Qualifying by Occurrence

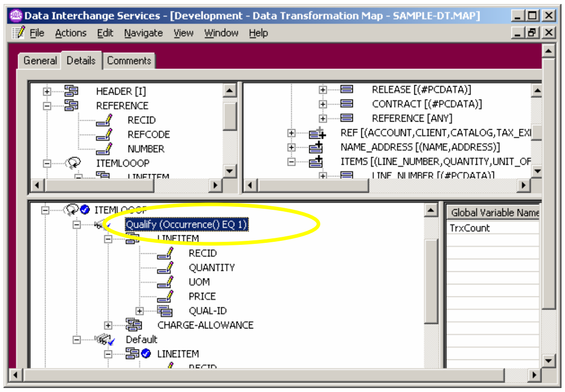


Qualify the segment by a specific occurrence of the segment.

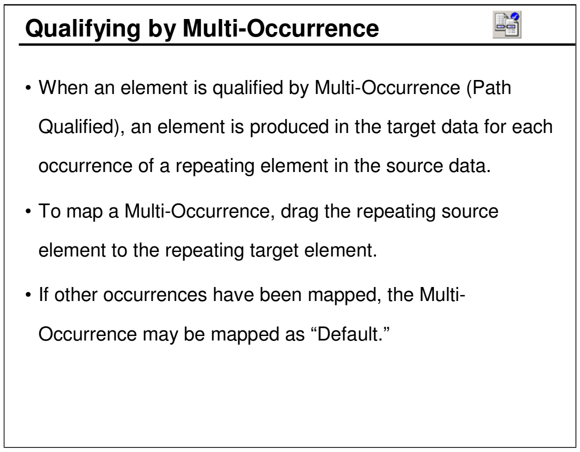
Enter the Occurrence Number: 1

This might be considered “positional” mapping. The first occurrence is mapped to one target, the second occurrence is mapped to another target, etc.

Occurrence Qualification



This is stating that when this source data appears for the first time, perform this mapping.

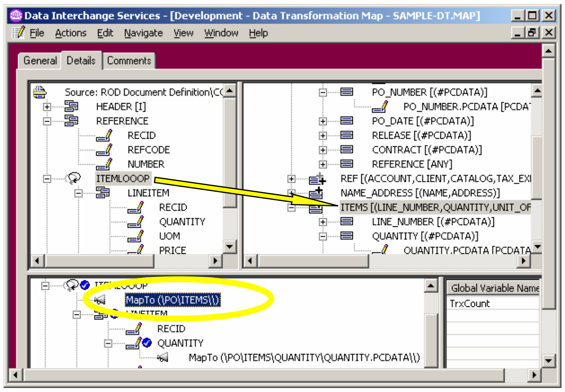


Qualifying by Multi-Occurrence

- When an element is qualified by Multi-Occurrence (Path Qualified), an element is produced in the target data for each occurrence of a repeating element in the source data.
- To map a Multi-Occurrence, drag the repeating source element to the repeating target element.
- If other occurrences have been mapped, the Multi-Occurrence may be mapped as “Default.”

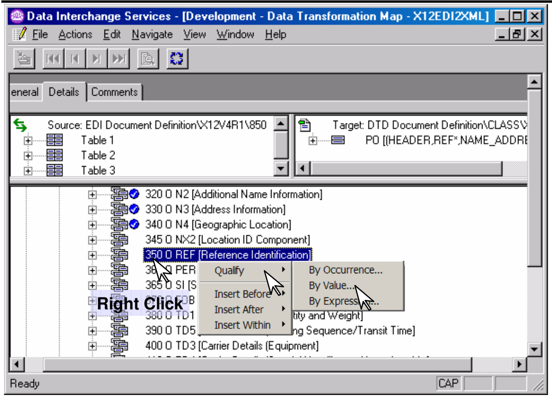
For a source-based map this will produce a *MapTo* command. For a target-based map, a *ForEach* command will be generated.

Multi-Occurrence MapTo



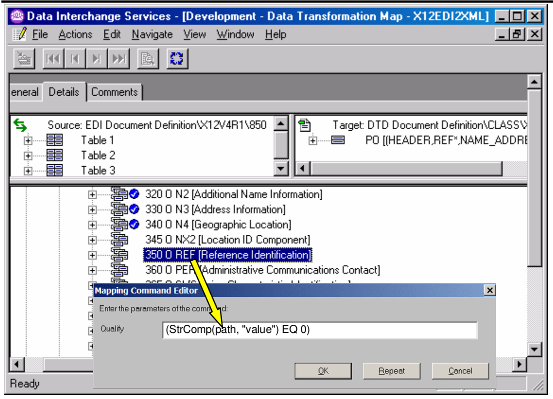
For each occurrence of ITEMLOOP in the source data, one ITEMS element in the XML target data will be created.

Qualifying by Value


Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]
File Actions Edit Navigate View Window Help
eneral
Details
Comments
Source: EDI Document Definition\X12V4R1\850
Table 1
Table 2
Table 3
Target: DTD Document Definition\CLASS\
PO [(HEADER,REF*,NAME_ADDRE
320 O N2 [Additional Name Information]
330 O N3 [Address Information]
340 O N4 [Geographic Location]
345 O NX2 [Location ID Component]
350 O REF [Reference Identification]
36_ O PER
365 O SI [S
_ O FOB
380 O TD1
tity and Weight]
390 O TD5
ng Sequence/Transit Time]
400 O TD3 [Carrier Details (Equipment]
Right Click
Qualify
Insert Before
Insert After
Insert Within
By Occurrence...
By Value...
By Express...
Ready
CAP

Mapping is based on the value of a qualifying element.

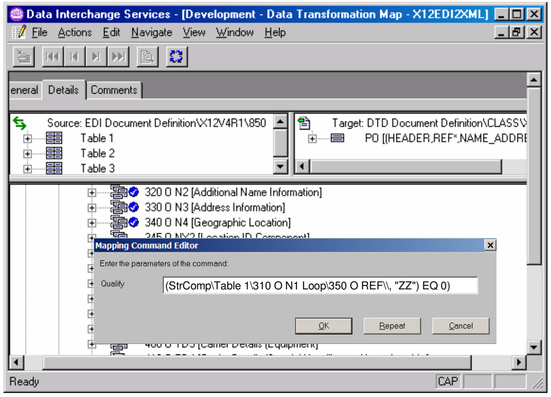
Qualifying by Value – Selecting the Path


Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]
File Actions Edit Navigate View Window Help
eneral Details Comments
Source: EDI Document Definition\X12V4R1\850
Table 1
Table 2
Table 3
Target: DTD Document Definition\CLASS\
PO [(HEADER,REF*,NAME_ADDRE
320 O N2 [Additional Name Information]
330 O N3 [Address Information]
340 O N4 [Geographic Location]
345 O NX2 [Location ID Component]
350 O REF [Reference Identification]
360 O PER [Administrative Communications Contact]
Mapping Command Editor
Enter the parameters of the command:
OK
Repeat
Cancel
Ready

Qualify: (StrComp(path, "value") EQ 0)

When comparing string values you need to use the StrComp (string comparison) function. If you drag element 128 to “path”, you can compare it a value that you enter. If the value of element 128 is equal to the literal value entered, the StrComp function will return a “True”. This will cause the mapping to occur.

Qualifying by Value - Completed

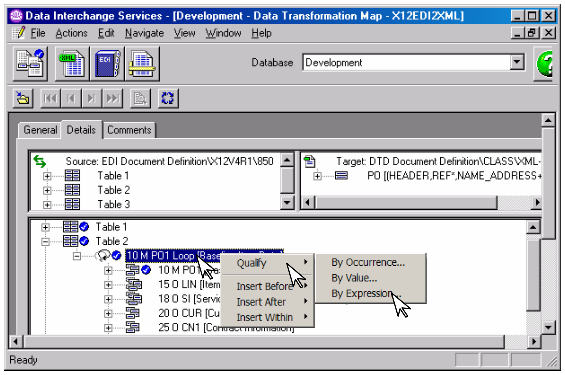

Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]
File
Actions
Edit
Navigate
View
Window
Help
eneral
Details
Comments
Source: EDI Document Definition\X12V4R1\850
Table 1
Table 2
Table 3
Target: DTD Document Definition\CLASS\
PO [(HEADER,REF*,NAME_ADDRE
320 O N2 [Additional Name Information]
330 O N3 [Address Information]
340 O N4 [Geographic Location]
Mapping Command Editor
OK
Repeat
Cancel
Ready
CAP

Enter the parameters of the command:

Qualify: (StrComp\Table 1\310 O N1 Loop\350 O REF\\, "ZZ") EQ 0)

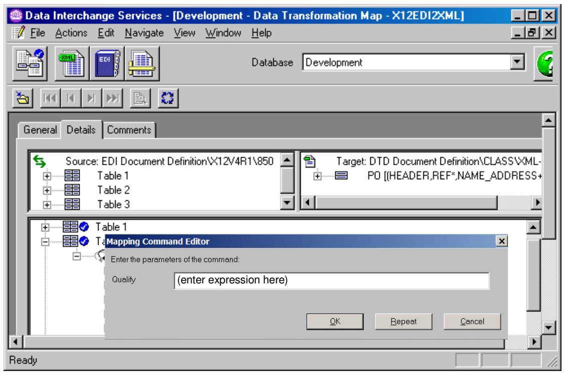
If element 128 contains ‘ZZ’ the StrComp (string comparison) will return a “True” and the mapping will take place.

Qualifying by Expression


Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]
File Actions Edit Navigate View Window Help
Database
Development
General Details Comments
Source: EDI Document Definition\X12V4R1\850
Table 1
Table 2
Table 3
Target: DTD Document Definition\CLASS\XML-
PO [(HEADER,REF*,NAME_ADDRESS+
Table 1
Table 2
10 M PO1 Loop [Base
10 M PO1
15 O LIN [Item
18 O SI [Servic
20 O CUR [Cu
25 O CN1 [Contract Information]
Qualify
Insert Before
Insert After
Insert Within
By Occurrence...
By Value...
By Expression...
Ready

Here the mapping will take place if a logical expression is true.

Defining a Qualifying Expression



Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]

Database: Development

Source: EDI Document Definition\X12V4R1\850

Target: DTD Document Definition\CLASS\XML-

PO [(HEADER,REF*,NAME_ADDRESS+

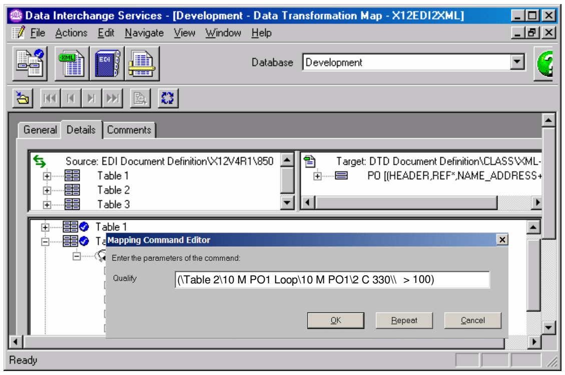
Mapping Command Editor

Enter the parameters of the command:

Qualify: (enter expression here)

Here you can enter any expression that evaluates to true or false.

Entering the Qualifying Expression



Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]
File Actions Edit Navigate View Window Help
Database
General Details Comments
Source: EDI Document Definition\X12V4R1\850
Table 1
Table 2
Table 3
Target: DTD Document Definition\CLASS\XML-
PO [(HEADER,REF*,NAME_ADDRESS+
Table 1
Ta
Mapping Command Editor
OK
Repeat
Cancel
Ready

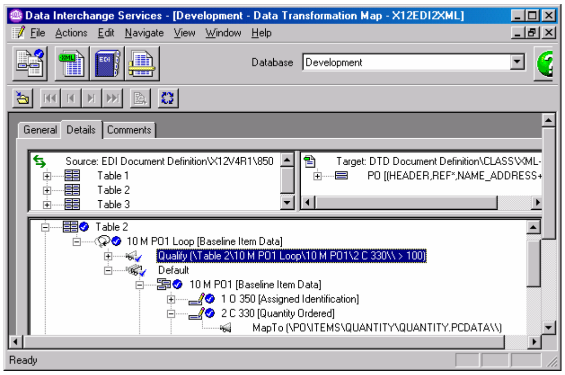
Database: Development

Enter the parameters of the command:

Qualify: (\Table 2\10 M PO1 Loop\10 M PO1\2 C 330\\ > 100)

If the field defined by path “\LINE_ITEMS\ITEMINFOS\QUANTITY\\” is greater than 100, the mapping will be performed.

Qualifying by Expression - Completed


Data Interchange Services - [Development - Data Transformation Map - X12EDI2XML]
File
Actions
Edit
Navigate
View
Window
Help
EDI
General
Details
Comments
Ready

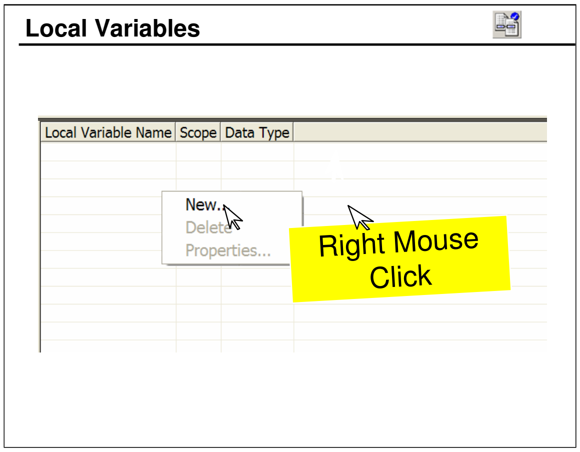
Database: Development

```
Source: EDI Document Definition\X12V4R1\850
  Table 1
  Table 2
  Table 3

Target: DTD Document Definition\CLASS\XML-
  PO [(HEADER,REF*,NAME_ADDRESS+

Table 2
  10 M PO1 Loop [Baseline Item Data]
    Qualify (\Table 2\10 M PO1 Loop\10 M PO1\2 C 330\\ > 100)
    Default
      10 M PO1 [Baseline Item Data]
        1 O 350 [Assigned Identification]
        2 C 330 [Quantity Ordered]
          MapTo (\PO\ITEMS\QUANTITY\QUANTITY.PCDATA\\)
```

The Default mapping will occur if none of the Qualifying conditions is 'True'.

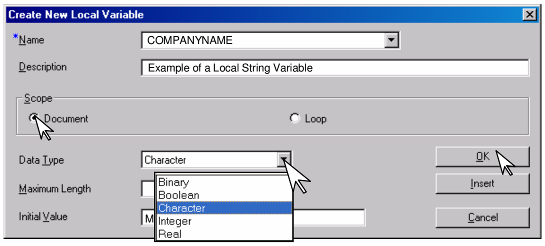


Local Variables

Right Mouse Click

Be sure to right click below the column header bar.

Defining a New Local Variable



Create New Local Variable

*Name: COMPANYNAME

Description: Example of a Local String Variable

Scope: ● Document ○ Loop

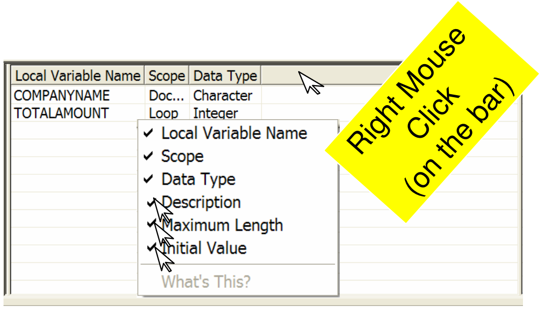
Data Type: Character

Maximum Length:

Initial Value: M

A local variable is one that exists for the duration of the mapping of the current Document Definition or a loop within the current Document Definition.

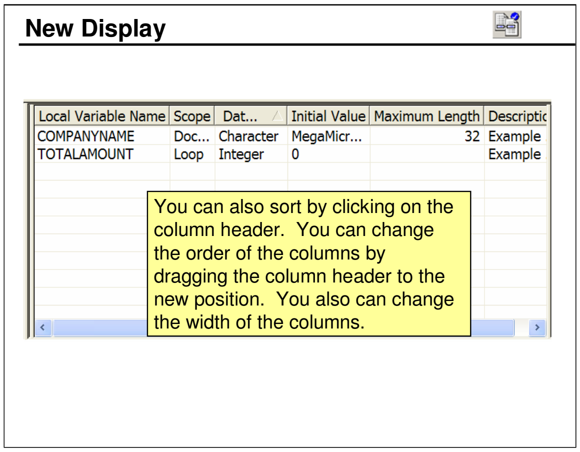
Selecting Display Columns



Local Variable Name	Scope	Data Type
COMPANYNAME	Doc...	Character
TOTALAMOUNT	Loop	Integer

Right Mouse Click (on the bar)

Right click in the white area to add a variable. Right click on the header bar to select display columns.



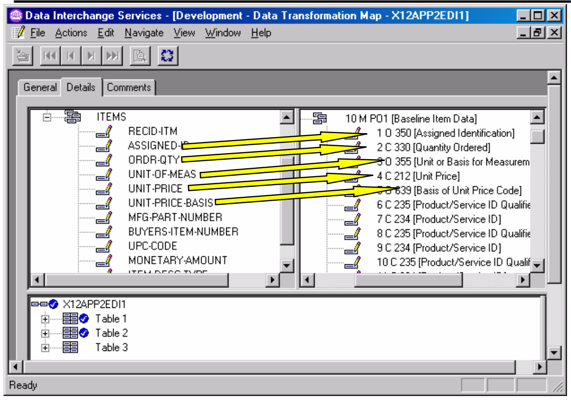
New Display

Local Variable Name	Scope	Dat...	Initial Value	Maximum Length	Descriptic
COMPANYNAME	Doc...	Character	MegaMicr...	32	Example
TOTALAMOUNT	Loop	Integer	0		Example

You can also sort by clicking on the column header. You can change the order of the columns by dragging the column header to the new position. You also can change the width of the columns.

This feature is not available for lists of objects within functional areas.

Drag and Drop Mapping – Target Based


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File Actions Edit Navigate View Window Help
General Details Comments
Ready

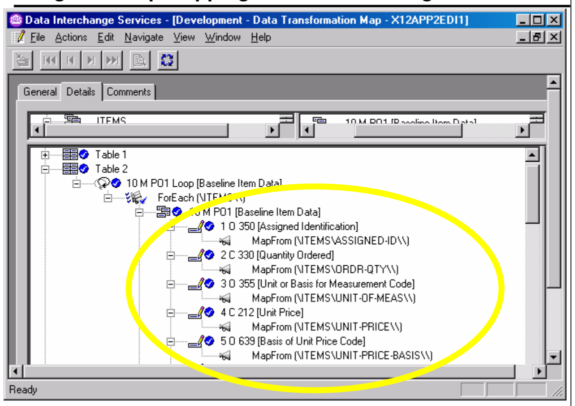
ITEMS
- RECID-ITM
- ASSIGNED-I
- ORDR-QTY
- UNIT-OF-MEAS
- UNIT-PRICE
- UNIT-PRICE-BASIS
- MFG-PART-NUMBER
- BUYERS-ITEM-NUMBER
- UPC-CODE
- MONETARY-AMOUNT

10 M PO1 [Baseline Item Data]
- 1 O 350 [Assigned Identification]
- 2 C 330 [Quantity Ordered]
- 3 O 355 [Unit or Basis for Measurem
- 4 C 212 [Unit Price]
- 5 O 639 [Basis of Unit Price Code]
- 6 C 235 [Product/Service ID Qualifie
- 7 C 234 [Product/Service ID]
- 8 C 235 [Product/Service ID Qualifie
- 9 C 234 [Product/Service ID]
- 10 C 235 [Product/Service ID Qualif

X12APP2EDI1
- Table 1
- Table 2
- Table 3

Note that the data need not appear in the same order in the source and target documents.

Drag and Drop Mapping Commands – Target Based

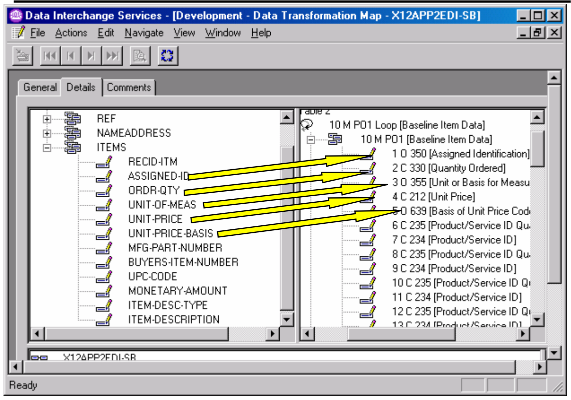


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File
Actions
Edit
Navigate
View
Window
Help
General
Details
Comments
ITEMS
10 M P01 [Baseline Item Data]
Ready

```
Table 1
Table 2
    10 M P01 Loop [Baseline Item Data]
        ForEach (\ITEMS\\)
            10 M P01 [Baseline Item Data]
                1 O 350 [Assigned Identification]
                    MapFrom (\ITEMS\ASSIGNED-ID\\)
                2 C 330 [Quantity Ordered]
                    MapFrom (\ITEMS\ORDR-QTY\\)
                3 O 355 [Unit or Basis for Measurement Code]
                    MapFrom (\ITEMS\UNIT-OF-MEAS\\)
                4 C 212 [Unit Price]
                    MapFrom (\ITEMS\UNIT-PRICE\\)
                5 O 639 [Basis of Unit Price Code]
                    MapFrom (\ITEMS\UNIT-PRICE-BASIS\\)
```

Note that *MapTo* commands are produced when performing drag-and-drop for source-based maps.

Drag and Drop Mapping – Source Based



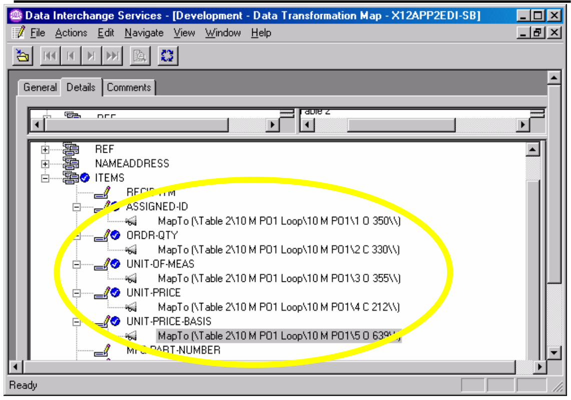
Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI-SB]
File
Actions
Edit
Navigate
View
Window
Help
General
Details
Comments
Ready

REF
NAMEADDRESS
ITEMS
RECID-ITM
ASSIGNED-ID
ORDR-QTY
UNIT-OF-MEAS
UNIT-PRICE
UNIT-PRICE-BASIS
MFG-PART-NUMBER
BUYERS-ITEM-NUMBER
UPC-CODE
MONETARY-AMOUNT
ITEM-DESC-TYPE
ITEM-DESCRIPTION

able 2
10 M PO1 Loop [Baseline Item Data]
10 M PO1 [Baseline Item Data]
1 O 350 [Assigned Identification]
2 C 330 [Quantity Ordered]
3 O 355 [Unit or Basis for Measu
4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code
6 C 235 [Product/Service ID Qu
7 C 234 [Product/Service ID]
8 C 235 [Product/Service ID Qu
9 C 234 [Product/Service ID]
10 C 235 [Product/Service ID Q
11 C 234 [Product/Service ID]
12 C 235 [Product/Service ID Q
13 C 234 [Product/Service ID]

X12APP2EDI-SB

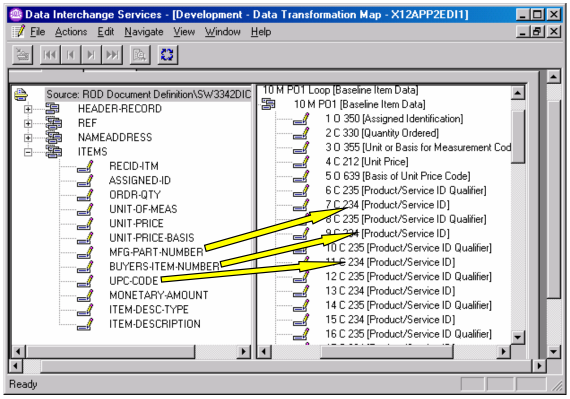
Drag and Drop Mapping Commands – Source Based


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI-SB]
File Actions Edit Navigate View Window Help
General Details Comments
REF
Table 2
Ready

```
REF
NAMEADDRESS
ITEMS
    RECIP ITM
    ASSIGNED-ID
        MapTo (\Table 2\10 M PO1 Loop\10 M PO1\1 O 350\\)
    ORDR-QTY
        MapTo (\Table 2\10 M PO1 Loop\10 M PO1\2 C 330\\)
    UNIT-OF-MEAS
        MapTo (\Table 2\10 M PO1 Loop\10 M PO1\3 O 355\\)
    UNIT-PRICE
        MapTo (\Table 2\10 M PO1 Loop\10 M PO1\4 C 212\\)
    UNIT-PRICE-BASIS
        MapTo (\Table 2\10 M PO1 Loop\10 M PO1\5 O 639\\)
    MFG-PART-NUMBER
```

Note that *MapFrom* commands are produced when performing drag-and-drop for target-based maps.

Mapping Qualified Elements - Outbound

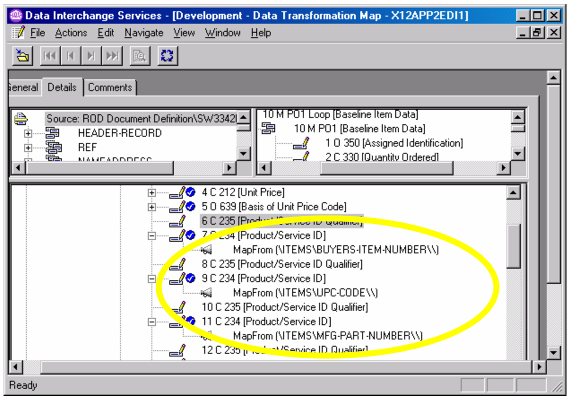

Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File Actions Edit Navigate View Window Help
Ready

```
Source: ROD Document Definition\SW3342DIC
  HEADER-RECORD
  REF
  NAMEADDRESS
  ITEMS
    RECID-ITM
    ASSIGNED-ID
    ORDR-QTY
    UNIT-OF-MEAS
    UNIT-PRICE
    UNIT-PRICE-BASIS
    MFG-PART-NUMBER
    BUYERS-ITEM-NUMBER
    UPC-CODE
    MONETARY-AMOUNT
    ITEM-DESC-TYPE
    ITEM-DESCRIPTION
```

```
10 M PO1 Loop [Baseline Item Data]
  10 M PO1 [Baseline Item Data]
    1 O 350 [Assigned Identification]
    2 C 330 [Quantity Ordered]
    3 O 355 [Unit or Basis for Measurement Cod
    4 C 212 [Unit Price]
    5 O 639 [Basis of Unit Price Code]
    6 C 235 [Product/Service ID Qualifier]
    7 C 234 [Product/Service ID]
    8 C 235 [Product/Service ID Qualifier]
    9 C 234 [Product/Service ID]
    10 C 235 [Product/Service ID Qualifier]
    11 C 234 [Product/Service ID]
    12 C 235 [Product/Service ID Qualifier]
    13 C 234 [Product/Service ID]
    14 C 235 [Product/Service ID Qualifier]
    15 C 234 [Product/Service ID]
    16 C 235 [Product/Service ID Qualifier]
```

Here the source data does not contain the qualifying values, but places the product IDs in uniquely named fields. All these fields need to be mapped to element number 234 and will need qualifying values to distinguish them in the EDI data.

Qualified Elements Mapped


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File
Actions
Edit
Navigate
View
Window
Help
General
Details
Comments
Ready

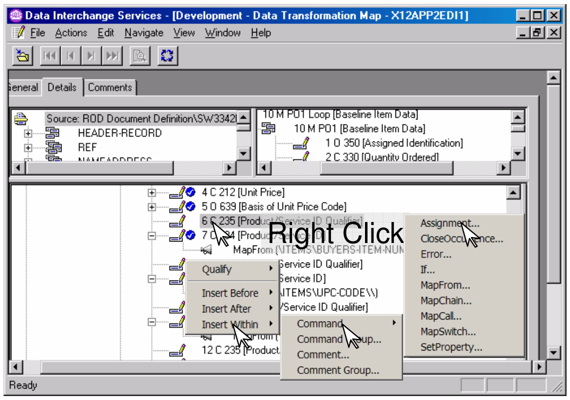
```
Source: ROD Document Definition\SW3342I
  HEADER-RECORD
  REF
  NAMEADDRESS

10 M PO1 Loop [Baseline Item Data]
  10 M PO1 [Baseline Item Data]
    1 O 350 [Assigned Identification]
    2 C 330 [Quantity Ordered]

4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
7 C 234 [Product/Service ID]
    MapFrom (\ITEMS\BUYERS-ITEM-NUMBER\\)
8 C 235 [Product/Service ID Qualifier]
9 C 234 [Product/Service ID]
    MapFrom (\ITEMS\UPC-CODE\\)
10 C 235 [Product/Service ID Qualifier]
11 C 234 [Product/Service ID]
    MapFrom (\ITEMS\MFG-PART-NUMBER\\)
12 C 235 [Product/Service ID Qualifier]
```

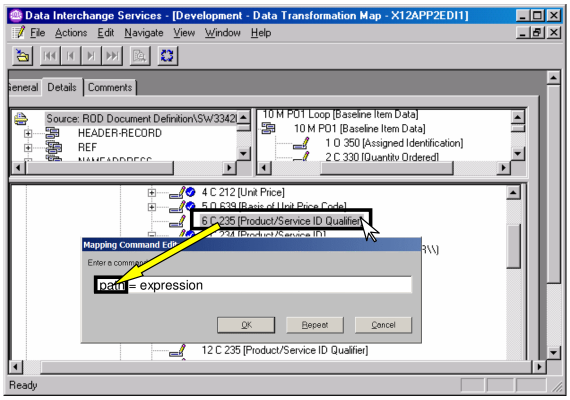
Element 234 was mapped from source data. The qualifiers in elements 235 will be manually hard-coded.

Mapping Literal Qualifiers – Insert / Command


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File Actions Edit Navigate View Window Help
General Details Comments
Source: ROD Document Definition\SW3342
HEADER-RECORD
REF
10 M PO1 Loop [Baseline Item Data]
10 M PO1 [Baseline Item Data]
1 O 350 [Assigned Identification]
2 C 330 [Quantity Ordered]
4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
7 C 234 [Product/Service ID]
MapFrom (\ITEMS\BUYERS-ITEM-NUM
Service ID Qualifier]
Service ID]
\ITEMS\UPC-CODE\\)
/Service ID Qualifier]
12 C 235 [Product
Right Click
Qualify
Insert Before
Insert After
Insert Within
Command
Command Group...
Comment...
Comment Group...
Assignment...
CloseOccurrence...
Error...
If...
MapFrom...
MapChain...
MapCall...
MapSwitch...
SetProperty...
Ready

Use assignment statements to establish the qualifying values, by setting the qualifiers to constant values.

Assigning the Qualifier to Element 235



Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]

Source: ROD Document Definition\SW3342I
HEADER-RECORD
REF

10 M PO1 Loop [Baseline Item Data]
10 M PO1 [Baseline Item Data]
1 O 350 [Assigned Identification]
2 C 330 [Quantity Ordered]

4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
12 C 235 [Product/Service ID Qualifier]

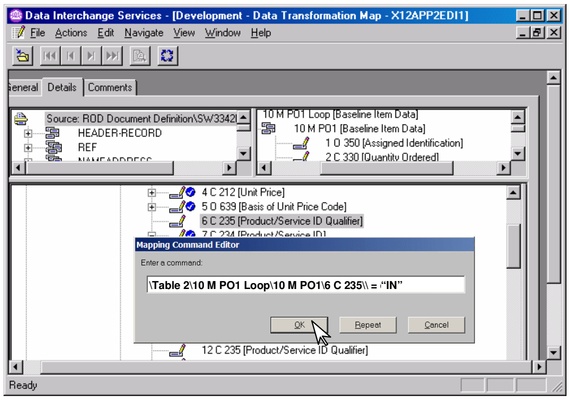
Mapping Command Edit

Enter a command

path = expression

Paths are always created by dragging from the source or target value being mapped.

Qualifier Assignment from a Literal


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File Actions Edit Navigate View Window Help
General Details Comments
OK
Repeat
Cancel
Ready

Source: ROD Document Definition\SW3342
HEADER-RECORD
REF
NAMEADDRESS

10 M PO1 Loop [Baseline Item Data]
10 M PO1 [Baseline Item Data]
1 O 350 [Assigned Identification]
2 C 330 [Quantity Ordered]

4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
7 C 234 [Product/Service ID]
12 C 235 [Product/Service ID Qualifier]

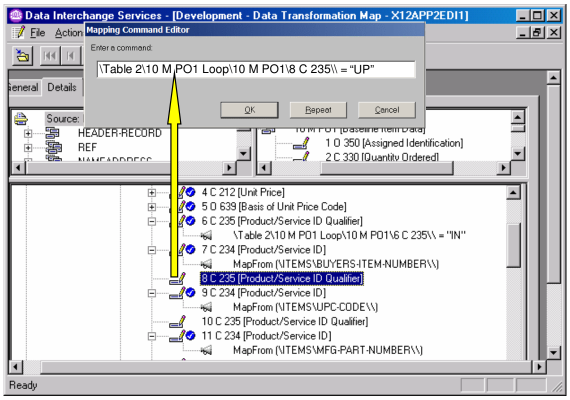
Mapping Command Editor

Enter a command:

```
\Table 2\10 M PO1 Loop\10 M PO1\6 C 235\\ = “IN”
```

Here one of the element 235 qualifiers is mapped to the literal ‘IN’ (Item Number).

Mapping the Next Qualifier



Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File
Action
Mapping Command Editor
General
Details
Source:
OK
Repeat
Cancel
Ready

Enter a command:

```
\Table 2\10 M PO1 Loop\10 M PO1\8 C 235\\ = “UP”
```

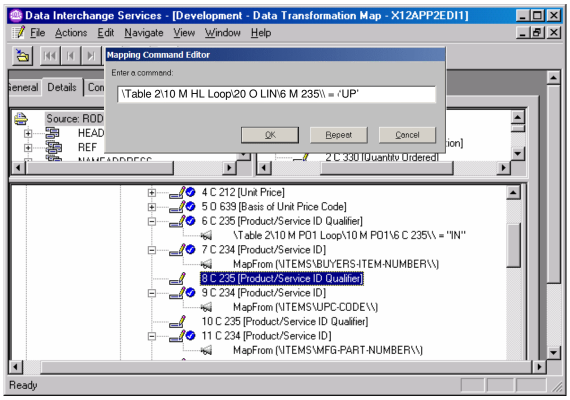
```
HEADER-RECORD
REF
```

```
1 O 350 [Assigned Identification]
2 C 330 [Quantity Ordered]
```

```
4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
    \Table 2\10 M PO1 Loop\10 M PO1\6 C 235\\ = "IN"
7 C 234 [Product/Service ID]
    MapFrom (\ITEMS\BUYERS-ITEM-NUMBER\\)
8 C 235 [Product/Service ID Qualifier]
9 C 234 [Product/Service ID]
    MapFrom (\ITEMS\UPC-CODE\\)
10 C 235 [Product/Service ID Qualifier]
11 C 234 [Product/Service ID]
    MapFrom (\ITEMS\MFG-PART-NUMBER\\)
```

In this slide, ‘UP’ (Qualifier for UPC Code) was mapped to element 235 to qualify the following element 234. Element 234 has been mapped the UPC code from the source data.

Mapping Qualifier “UP” as a Literal



Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]

Mapping Command Editor

Enter a command:

```
\Table 2\10 M HL Loop\20 O LIN\6 M 235\\ = ‘UP’
```

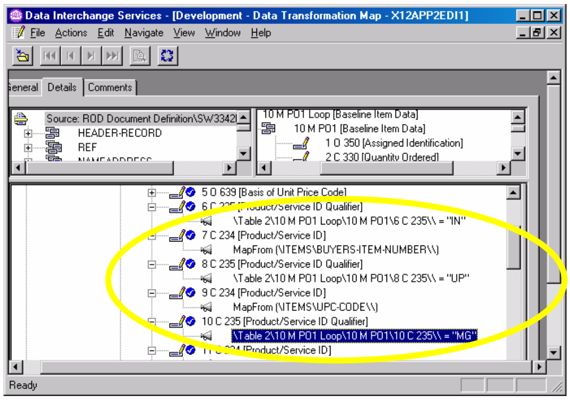
```
Source: ROD
  HEAD
  REF
  NAMEADDRESS
ion]
2 C 330 [Quantity Ordered]
```

```
4 C 212 [Unit Price]
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
      \Table 2\10 M PO1 Loop\10 M PO1\6 C 235\\ = "IN"
7 C 234 [Product/Service ID]
      MapFrom (\ITEMS\BUYERS-ITEM-NUMBER\\)
8 C 235 [Product/Service ID Qualifier]
9 C 234 [Product/Service ID]
      MapFrom (\ITEMS\UPC-CODE\\)
10 C 235 [Product/Service ID Qualifier]
11 C 234 [Product/Service ID]
      MapFrom (\ITEMS\MFG-PART-NUMBER\\)
```

Ready

Finally ‘UP’ (UPC code) is mapped to qualify the element containing the UPC code.

Completed Qualifier Mappings


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File Actions Edit Navigate View Window Help
General Details Comments
Ready

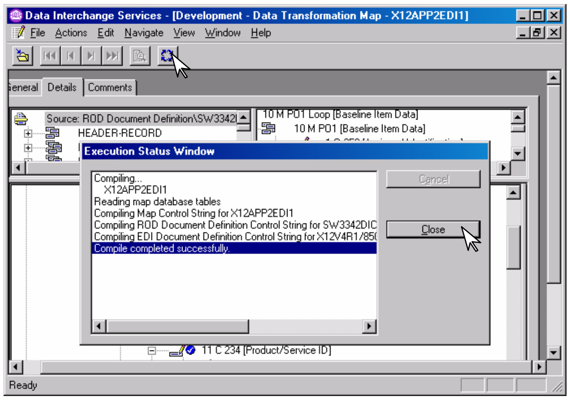
Source: ROD Document Definition\SW3342I
HEADER-RECORD
REF
NAMEADDRESS

10 M PO1 Loop [Baseline Item Data]
10 M PO1 [Baseline Item Data]
1 O 350 [Assigned Identification]
2 C 330 [Quantity Ordered]

```
5 O 639 [Basis of Unit Price Code]
6 C 235 [Product/Service ID Qualifier]
        \Table 2\10 M PO1 Loop\10 M PO1\6 C 235\\ = "IN"
7 C 234 [Product/Service ID]
        MapFrom (\ITEMS\BUYERS-ITEM-NUMBER\\)
8 C 235 [Product/Service ID Qualifier]
        \Table 2\10 M PO1 Loop\10 M PO1\8 C 235\\ = "UP"
9 C 234 [Product/Service ID]
        MapFrom (\ITEMS\UPC-CODE\\)
10 C 235 [Product/Service ID Qualifier]
        \Table 2\10 M PO1 Loop\10 M PO1\10 C 235\\ = "MG"
11 C 234 [Product/Service ID]
```

It is important here to note the element positions in the paths since they appear similar.

Compile Control String

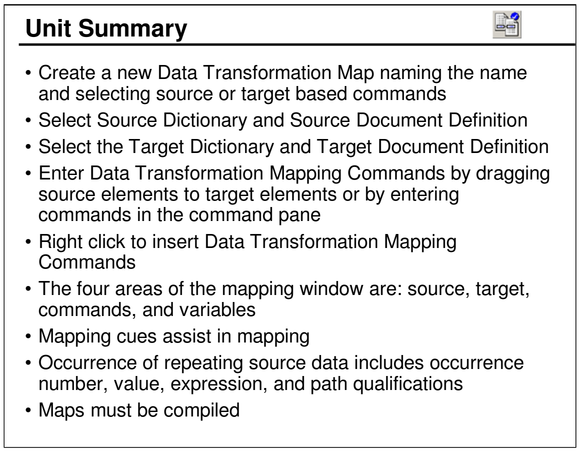


Data Interchange Services - [Development - Data Transformation Map - X12APP2EDI1]
File
Actions
Edit
Navigate
View
Window
Help
General
Details
Comments
Source: ROD Document Definition\SW3342
HEADER-RECORD
10 M PO1 Loop [Baseline Item Data]
10 M PO1 [Baseline Item Data]
Execution Status Window
Cancel
Close
11 C 234 [Product/Service ID]
Ready

```
Compiling...
   X12APP2EDI1
Reading map database tables
Compiling Map Control String for X12APP2EDI1
Compiling ROD Document Definition Control String for SW3342DIC
Compiling EDI Document Definition Control String for X12V4R1/850
Compile completed successfully.
```

If errors occur during compilation, check them carefully and correct the map. A successful compilation is essential.

Only Control Strings are used during translation. A change to a map is not effective until the Control String is compiled. The Control String may also need to be migrated to another database. The connection of remote Clients to the same database maybe accomplished through ODBC (open database connectivity) definitions.



Unit Summary

- Create a new Data Transformation Map naming the name and selecting source or target based commands
- Select Source Dictionary and Source Document Definition
- Select the Target Dictionary and Target Document Definition
- Enter Data Transformation Mapping Commands by dragging source elements to target elements or by entering commands in the command pane
- Right click to insert Data Transformation Mapping Commands
- The four areas of the mapping window are: source, target, commands, and variables
- Mapping cues assist in mapping
- Occurrence of repeating source data includes occurrence number, value, expression, and path qualifications
- Maps must be compiled

Each of these objectives have been discussed in this module.