



IBM Software Group

2004 WDI / WBIC Customer Conference

Global Business Transformation

What changes will I see moving to 3.2/Client
Configuration

Doug Hillary

IBM

WebSphere. software



 e-business software

© 2004 IBM Corporation

Objectives

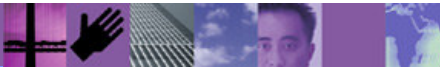
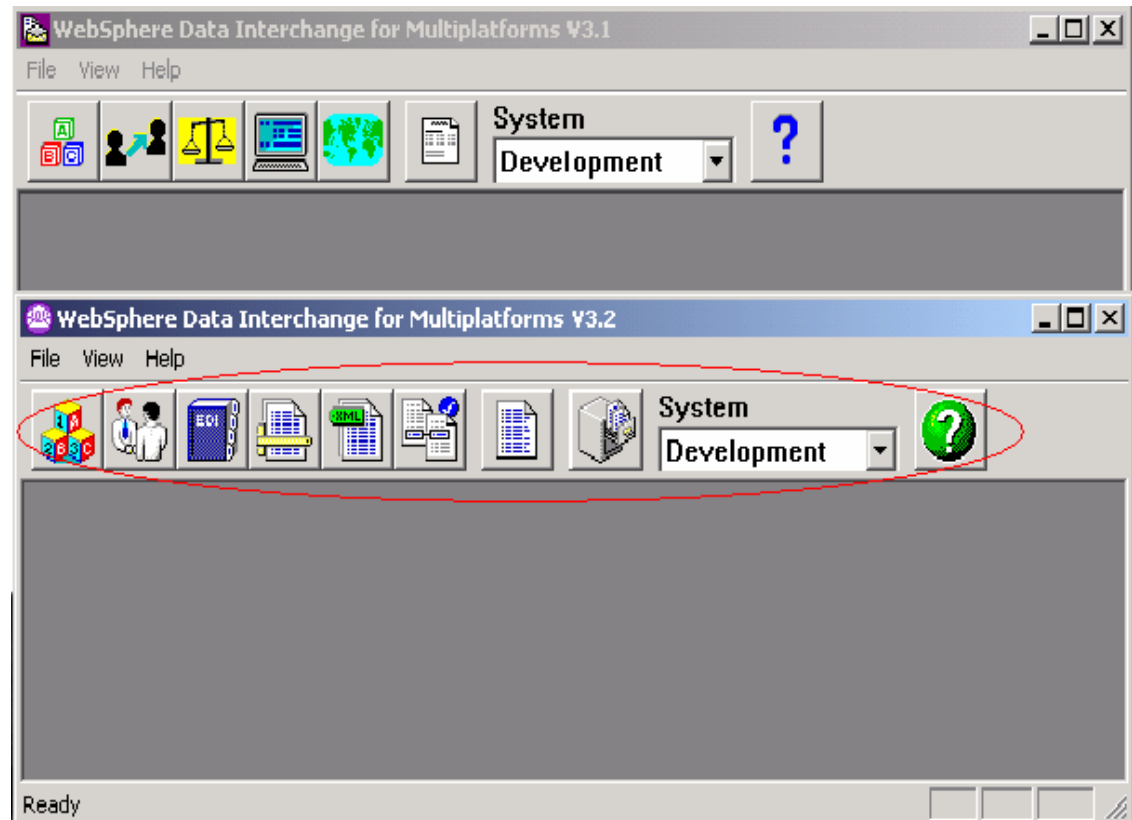
- Discuss the differences seen between WDI v3.1 Client and WDI v3.2 Client
- Discuss a brief overview of the Data Transformation mapper functionality
- Discuss added functions and features found in WDI v3.2



New Buttons

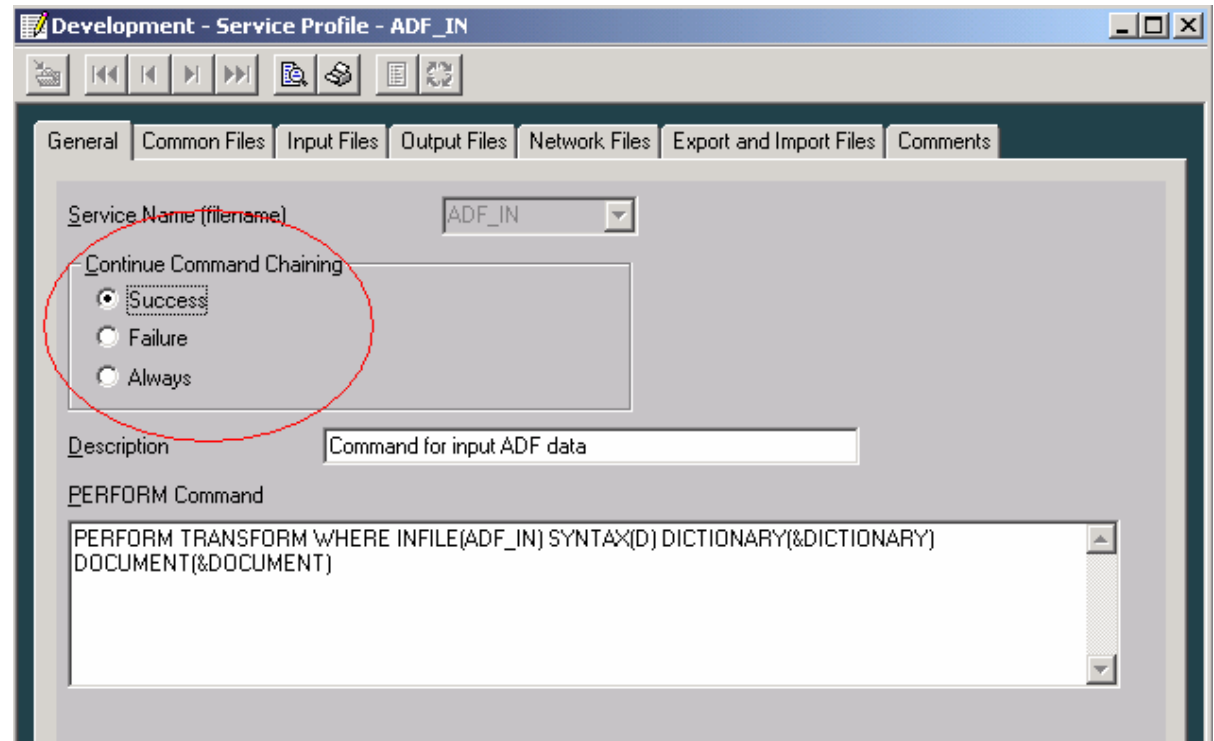
- Setup, Trading Partners, Standards, Data Formats, Mapping, and Transaction Store buttons look different.

- XML, and Rules buttons are new



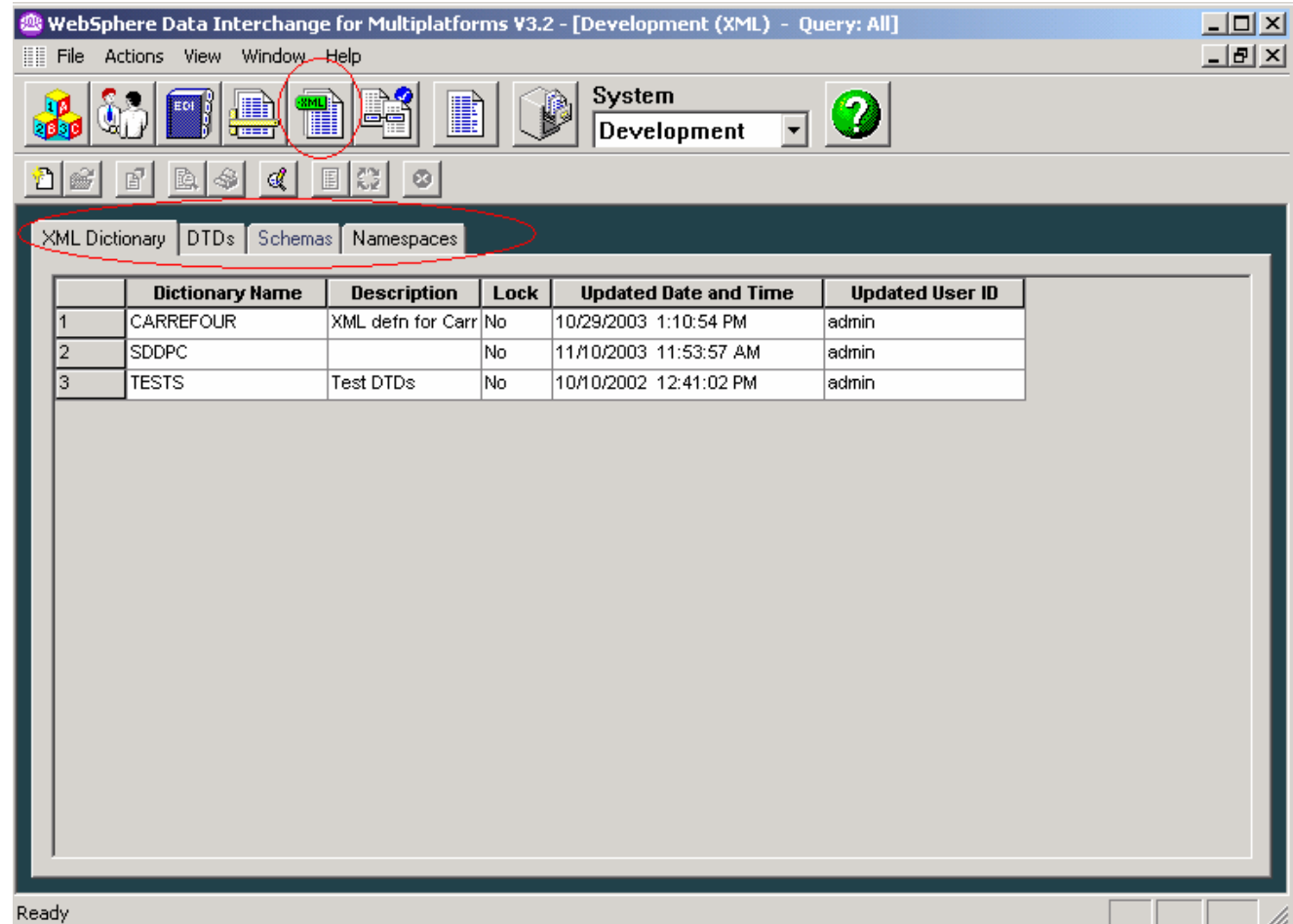
Service Profiles

- New options on Service Profiles allow you to specify if a command should be executed on Success, Failure, or Anytime the previous command completes



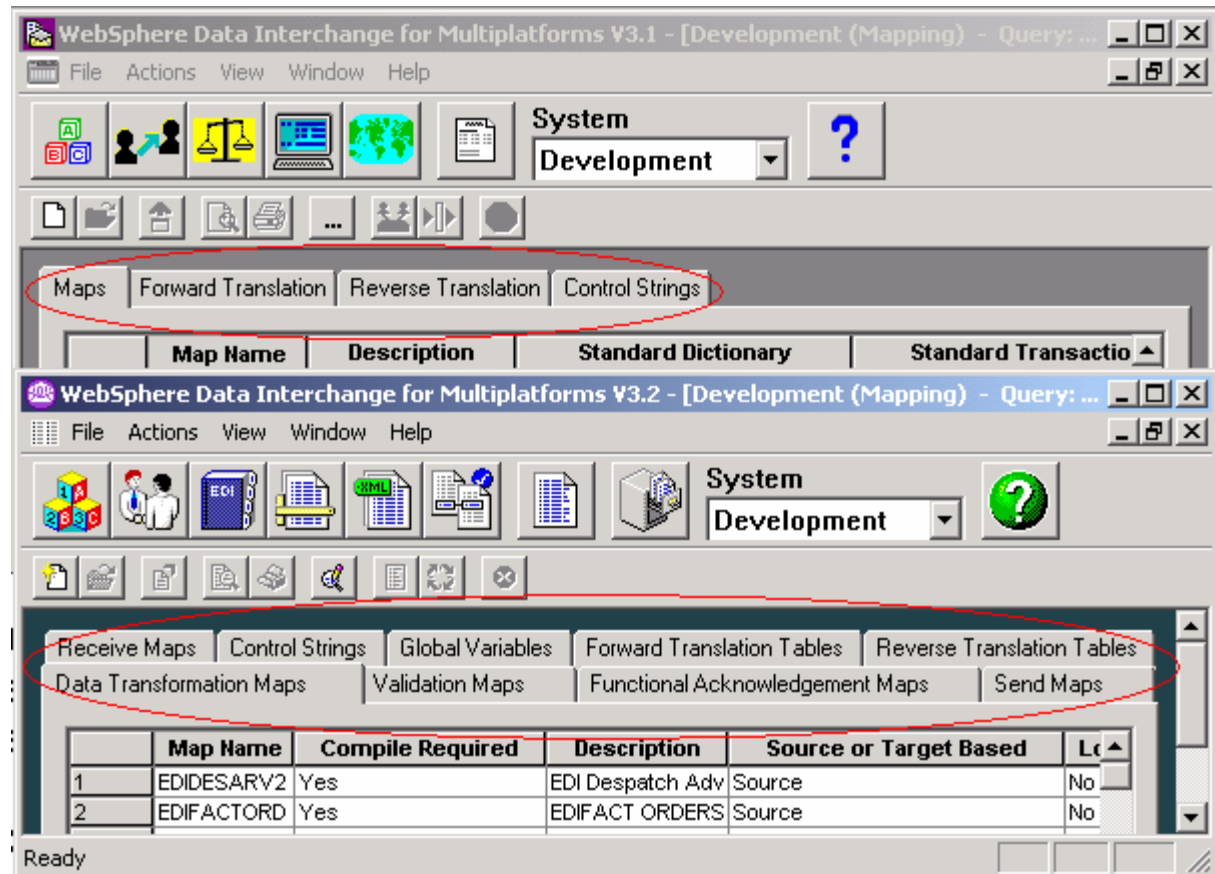
XML

- XML functional area allows you to maintain your XML objects the same as you would your Standards and Data Formats



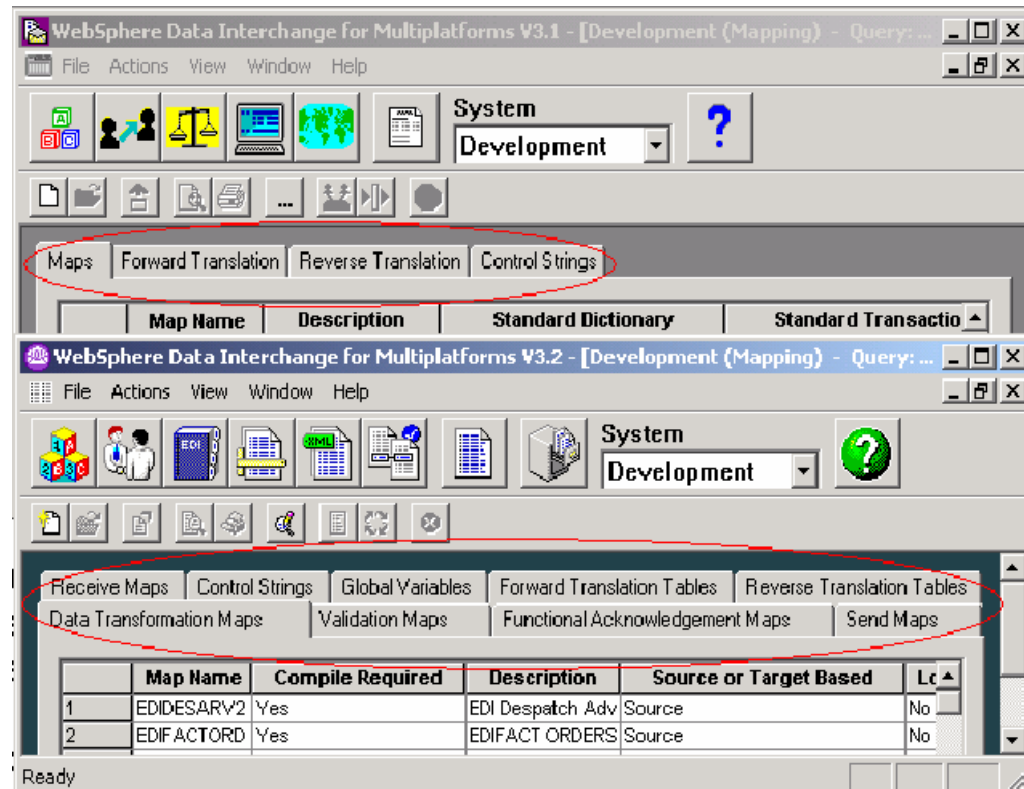
Mapping

Mapping tabs have been added:
 Maps have been broken into Send or Receive maps and can be used for maintenance of old S/R maps



Mapping

Global Variables, Validation Maps, and Functional Acknowledgement maps have been added to the mapping area.

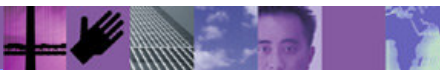


Mapping

The screenshot shows the WebSphere Data Interchange for Multiplatforms V3.2 - [Development (Mapping) - Query: All] window. The 'Global Variables' tab is selected and circled in red. Below the tabs is a table listing mapping information.

	Map Name	Compile Required	Description	Source or Target Based	Lock	Updated Date and Time	
1	EDIDESARV2	Yes	EDI Despatch Adv	Source	No	10/29/2003 1:09:36 PM	€
2	EDIFACTORD	Yes	EDIFACT ORDERS	Source	No	10/29/2003 1:10:04 PM	€
3	M191PARSER	Yes	m192	Target	No	10/29/2003 1:07:47 PM	€

- Global variables allows you to maintain variables without opening the maps. Global variables can now be identified as session, interchange, or group level variables



Mapping

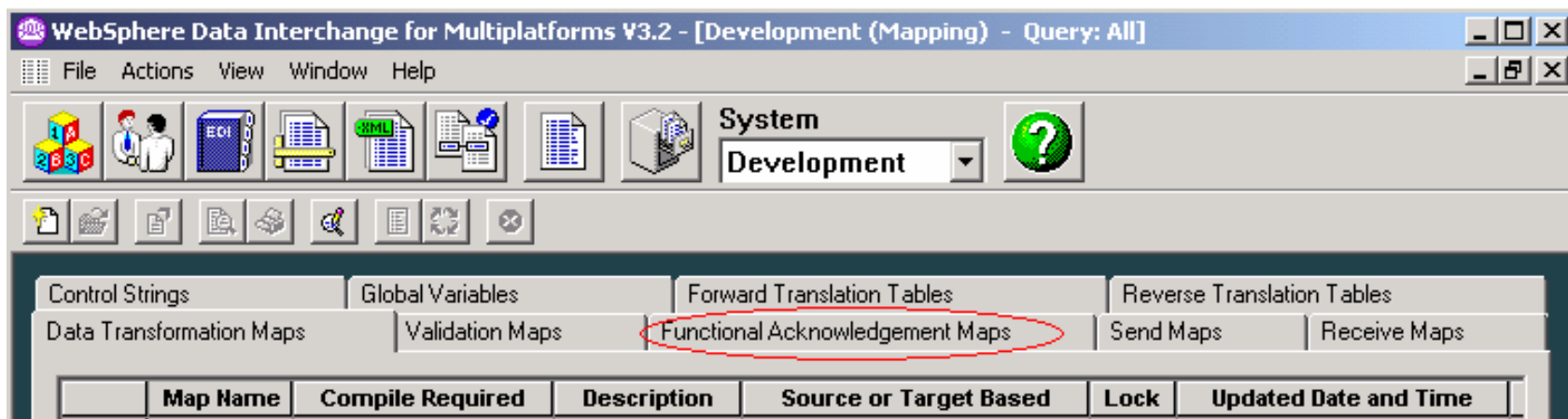
- Validation maps were added to provide implementation guide specific edits, allowing for you to respond as needed in the appropriate functional acknowledgement

	Map Name	Compile Required	Description	Source or Target Based	Lock	Updated Date and Time	
1	EDIDESARV2	Yes	EDI Despatch Adv	Source	No	10/29/2003 1:09:36 PM	6
2	EDIFACTORD	Yes	EDIFACT ORDERS	Source	No	10/29/2003 1:10:04 PM	6



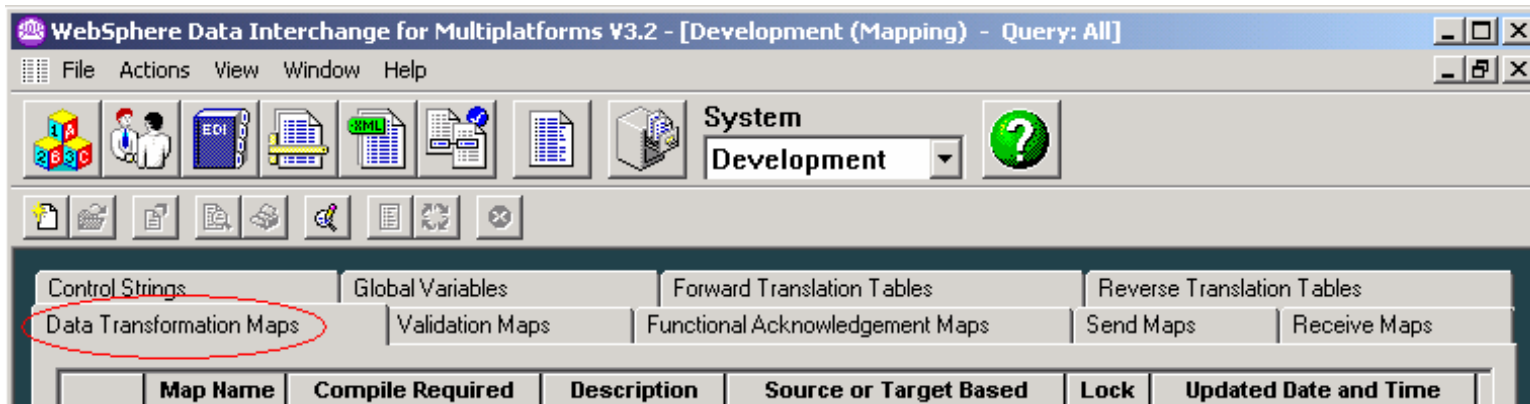
Mapping

- Functional Acknowledgement Maps allow you to edit the appropriate functional acknowledgement transaction if non standard processing is needed.



DT Maps

- Data Transformation maps offer exciting new mapping functions and useability features.



DT Maps

- Four panes – Source, Target, Mapping Commands, and Variables

The screenshot displays the WebSphere Data Interchange for Multiplatforms V3.2 interface for a Data Transformation Map (DT Map) named POXML55R-EDI. The interface is divided into four main panes:

- Source:** Shows the source XML structure, including elements like OrderSR, Header, PONum, PODate, Receiver, DetailLoop, ItemNumber, and SubDetail.
- Target:** Shows the target XML structure, including elements like Table 1, 20 M BEG, 40 O CUR, 50 O REF, 60 O PER, 70 O TAX, 80 O FC, 90 O CTP, 95 O PAM, 110 O CSH, and 115 O TC2.
- Mapping Commands:** Shows the mapping logic between source and target elements, such as MapTo and MapFrom commands.
- Variables:** Shows a table of global and local variables used in the mapping process.

Global Variable Name	Sc	Local Variable Name	S	Special Variable
global1	Se	ItemCount	D	DIOutType DIOutFile DICUserData

DT Maps

- Drag and drop mapping

The screenshot displays the WebSphere Data Interchange for Multiplatforms V3.2 interface for a Data Transformation Map (DT Map) titled "POXML5SR-EDI".

Source Structure (Left):

- OrderSR [(Header,DetailLoop*,Trailer)]
 - Header [(PONum,PODate,Sender,Receiver)]
 - Header.ATTLIST
 - PONum [(#PCDATA)]
 - PONum.PCDATA [PCDATA]
 - PODate [(#PCDATA)]
 - Sender [(Id,Qualifier)]
 - Receiver [(Id,Qualifier)]
 - DetailLoop [(ItemNumber,SubDetail+)]
 - ItemNumber [(#PCDATA)]
 - SubDetail [(Description,Quantity,UnitPrice)]

Target Structure (Right):

- 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]
 - 6 O 367 [Contract Number]
 - 7 O 587 [Acknowledgment Type]
 - 8 O 1019 [Invoice Type Code]
 - 9 O 1166 [Contract Type Code]
 - 10 O 1232 [Purchase Category]

Mapping: A mapping arrow points from the source element `PONum.PCDATA [PCDATA]` to the target element `324 [Purchase Order Number]`.

DT Map Editor (Bottom Left):

- OrderSR [(Header,DetailLoop*,Trailer)]
 - Header [(PONum,PODate,Sender,Receiver)]
 - Header.ATTLIST
 - \Table 1\{20 M BEG\2 M 92\} = "NE"
 - PONum [(#PCDATA)]
 - PONum.PCDATA [PCDATA]
 - MapTo (\Table 1\{20 M BEG\3 M 324\})
 - PODate [(#PCDATA)]
 - Sender [(Id,Qualifier)]
 - Receiver [(Id,Qualifier)]
 - DetailLoop [(ItemNumber,SubDetail+)]
 - MapTo (\Table 2\{10 M PO1 Loop\})
 - \Table 2\{10 M PO1 Loop\10 M PO1\1 O 350\} = ItemCount
 - ItemCount = ItemCount + 1

Variable Mapping Table (Bottom Right):

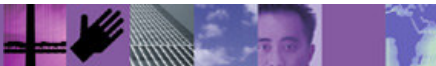
Global Variable Name	Sc	Local Variable Name	S	Special Variable I
global1	Se	ItemCount	D	DIOFileType DIOOutFile DICUserData

Commands between positions

- Positional mapping commands

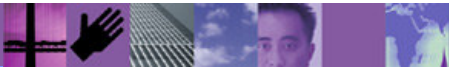
The screenshot displays the XML Editor interface with the following components:

- Source Tree (Left):** Shows the source XML structure for 'D:\TESTS\POXML5SR'. The 'Header.ATTLIST' element is selected, and a context menu is open over it. The 'Insert Within' option is highlighted, and its sub-menu is also open, showing 'Command' as the selected option.
- Target Tree (Right):** Shows the target XML structure for '20 M BEG [Beginning Segment for Purchase Order]'. Elements include '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]', '6 O 367 [Contract Number]', '7 O 587 [Acknowledgment Type]', '8 O 1019 [Invoice Type Code]', '9 O 1166 [Contract Type Code]', and '10 O 1232 [Purchase Category]'.
- Mapping Table (Bottom Right):** A table with columns for 'Global Variable Name', 'Local Variable Name', and 'Special Variable'. The first row contains 'global1', 'ItemCount', and 'DIOutType', 'DIOutFile', and 'DICUserData'.



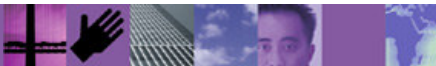
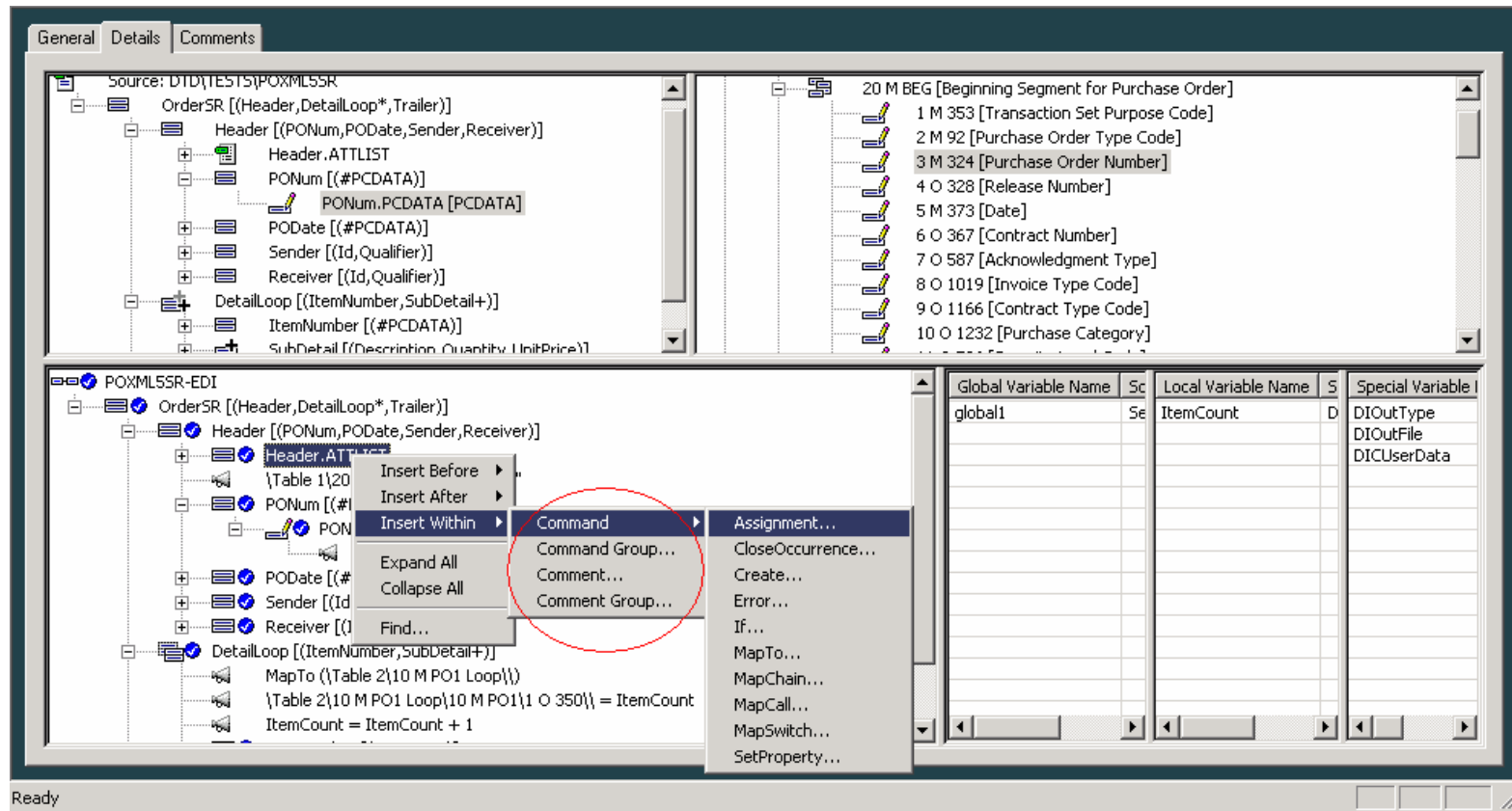
Commands between positions

- Inserting commands between segments or elements force an operation or assignment based on a specific location of processing data.
- Commands will be executed if the segment is present or not.
- Higher level Segments or Loops must be available to process the commands.



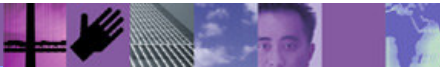
Groups of commands or comments

- Grouping options



Comments

- Ability to add comments to mapping commands.
- Ability to add comment groups to maps.
- Ability to add comments to the map overall
- Change control



Mapping commands

- More Mapping commands available

The screenshot displays the IBM WebSphere Mapping Tool interface. The top pane shows the source XML structure for 'Source: DTD\TESTS\POXML55R', including elements like 'OrderSR', 'Header', 'PONum', 'PODate', 'Sender', 'Receiver', 'DetailLoop', 'ItemNumber', and 'SubDetail'. The right pane shows the target XML structure for '20 M BEG [Beginning Segment for Purchase Order]', with elements like '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]', '6 O 367 [Contract Number]', '7 O 587 [Acknowledgment Type]', '8 O 1019 [Invoice Type Code]', '9 O 1166 [Contract Type Code]', and '10 O 1232 [Purchase Category]'. The bottom pane shows the mapping configuration for 'POXML55R-EDI', with a context menu open over the 'PONum' element. The 'Command' sub-menu is expanded, highlighting 'Assignment...'. Other visible mapping commands include 'CloseOccurrence...', 'Create...', 'Error...', 'If...', 'MapTo...', 'MapChain...', 'MapCall...', 'MapSwitch...', and 'SetProperty...'. A table on the right side of the bottom pane lists variables: Global Variable Name (global1), Local Variable Name (ItemCount), and Special Variable Name (DIOutType, DIOutFile, DICUserData).

Global Variable Name	Sc	Local Variable Name	S	Special Variable I
global1	Se	ItemCount	D	DIOutType
				DIOutFile
				DICUserData

Mapping commands

- RFH2 header and envelope properties are available in the map
 - GetProperty
 - SetProperty
- Map Chining available
 - MAPSWITCH – switch maps immediately
 - MAPCHAIN – reprocess input through second map after completion
 - MAPCALL – send a specific field to another “sub” map



Other Property values

Target Document Properties ▶	
Generic Envelope Properties ▶	"IchgCtlNum"
Specific Envelope Properties ▶	"IchgSndrQl"
MQMD Properties ▶	"IchgSndrId"
MQRFH2 Properties ▶	"IchgRcvrQl"
	"IchgRcvrId"
	"IchgDate"
	"IchgTime"
	"IchgPswd"
	"IchgUsgInd"
	"IchgAppRef"
	"IchgVerRel"
	"IchgGrpCnt"
	"IchgCtlTotal"
	"IchgTrxCnt"
	"GrpCtlNum"
	"GrpFuncGrpId"
	"GrpAppSndrId"
	"GrpAppRcvrId"
	"GrpDate"
	"GrpTime"
	"GrpPswd"
	"GrpVer"
	"GrpRel"
	"GrpTrxCnt"
	"TrxCtlNum"
	"TrxCd"
	"TrxVer"
	"TrxRel"
	"TrxSegCnt"

Target Document Properties ▶	
Generic Envelope Properties ▶	
Specific Envelope Properties ▶	"ISAnn" ("nn" is 01-16)
MQMD Properties ▶	"GSnn" ("nn" is 01-08)
MQRFH2 Properties ▶	"STnn" ("nn" is 01-02)
	"SEnn" ("nn" is 01-02)
	"GEnn" ("nn" is 01-02)
	"IEAnn" ("nn" is 01-02)
	"UNBnn" ("nn" is 01-18)
	"UNGnn" ("nn" is 01-13)
	"UNHnn" ("nn" is 01-09)
	"UNTnn" ("nn" is 01-02)
	"UNEnn" ("nn" is 01-02)
	"UNZnn" ("nn" is 01-02)
	"BGnn" ("nn" is 01-07)
	"EGnn" ("nn" is 01-04)



Other Property values

- Target Document Properties ▶
- Generic Envelope Properties ▶
- Specific Envelope Properties ▶
- MQMD Properties ▶**
- MQRFH2 Properties ▶

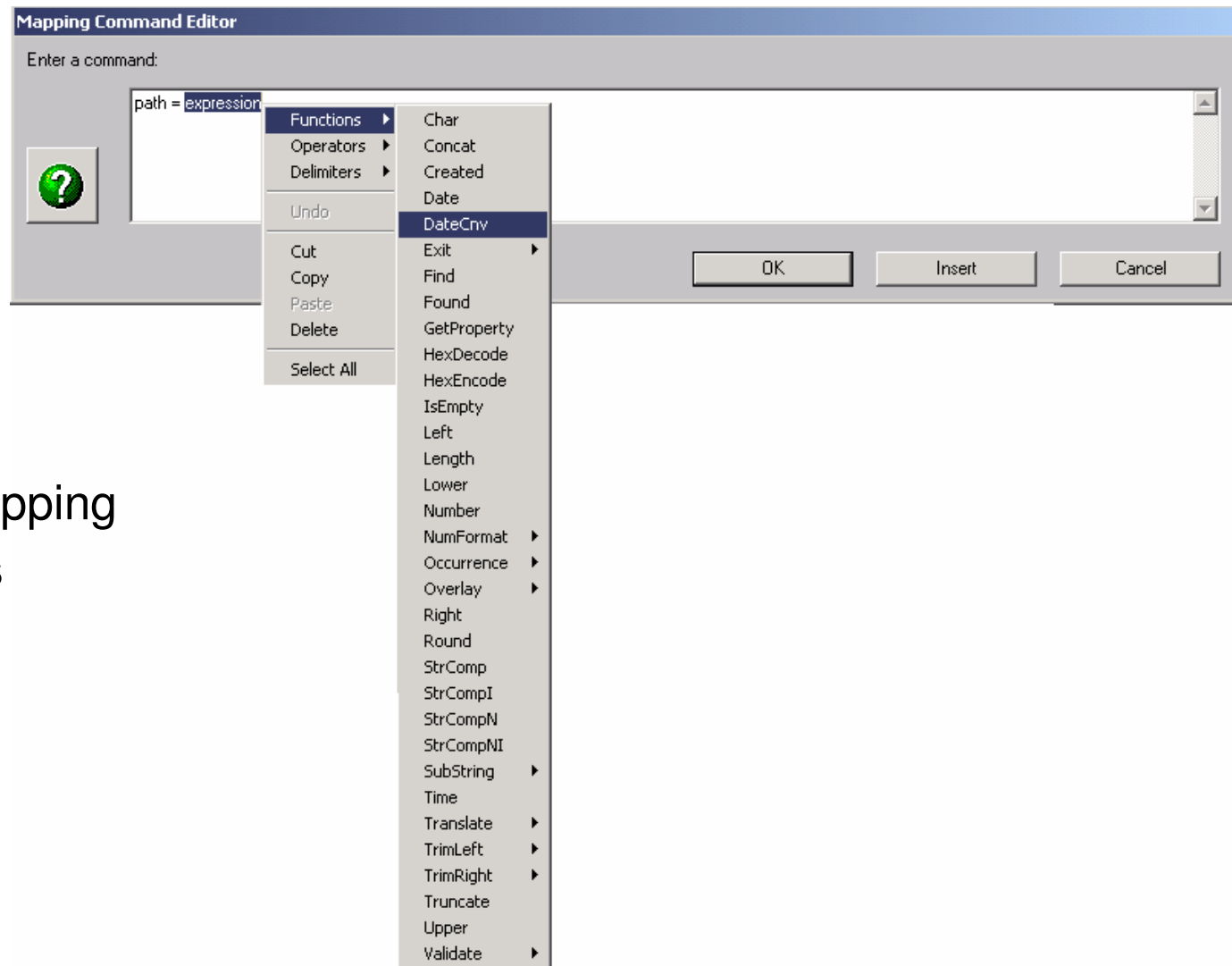
- "ROOT.MQMD.StrucId"
- "ROOT.MQMD.Version"
- "ROOT.MQMD.Report"
- "ROOT.MQMD.MsgType"
- "ROOT.MQMD.Expiry"
- "ROOT.MQMD.Feedback"
- "ROOT.MQMD.Encoding"
- "ROOT.MQMD.CodedCharSetId"
- "ROOT.MQMD.Format"
- "ROOT.MQMD.Priority"
- "ROOT.MQMD.Persistence"
- "ROOT.MQMD.MsgId"
- "ROOT.MQMD.CorrelId"
- "ROOT.MQMD.BackoutCount"
- "ROOT.MQMD.ReplyToQ"
- "ROOT.MQMD.ReplyToQMgr"
- "ROOT.MQMD.UserIdentifier"
- "ROOT.MQMD.AccountingToken"
- "ROOT.MQMD.AppIdentityData"
- "ROOT.MQMD.PutApplType"
- "ROOT.MQMD.PutApplName"
- "ROOT.MQMD.PutDate"
- "ROOT.MQMD.PutTime"
- "ROOT.MQMD.AppOriginData"
- "ROOT.MQMD.GroupId"
- "ROOT.MQMD.MsgSeqNumber"
- "ROOT.MQMD.Offset"
- "ROOT.MQMD.MsgFlags"
- "ROOT.MQMD.OriginalLength"

- Target Document Properties ▶
- Generic Envelope Properties ▶
- Specific Envelope Properties ▶
- MQMD Properties ▶
- MQRFH2 Properties ▶**

- "ROOT.MQRFH2.StrucId"
- "ROOT.MQRFH2.Version"
- "ROOT.MQRFH2.StrucLength"
- "ROOT.MQRFH2.Encoding"
- "ROOT.MQRFH2.CodedCharSetId"
- "ROOT.MQRFH2.Format"
- "ROOT.MQRFH2.Flags"
- "ROOT.MQRFH2.NameValueCCSID"
- "ROOT.MQRFH2.mcd.Msd"
- "ROOT.MQRFH2.mcd.Set"
- "ROOT.MQRFH2.mcd.Type"
- "ROOT.MQRFH2.mcd.Fmt"
- "ROOT.MQRFH2.usr.xxxxxx"



Mapping functions



- More mapping functions available

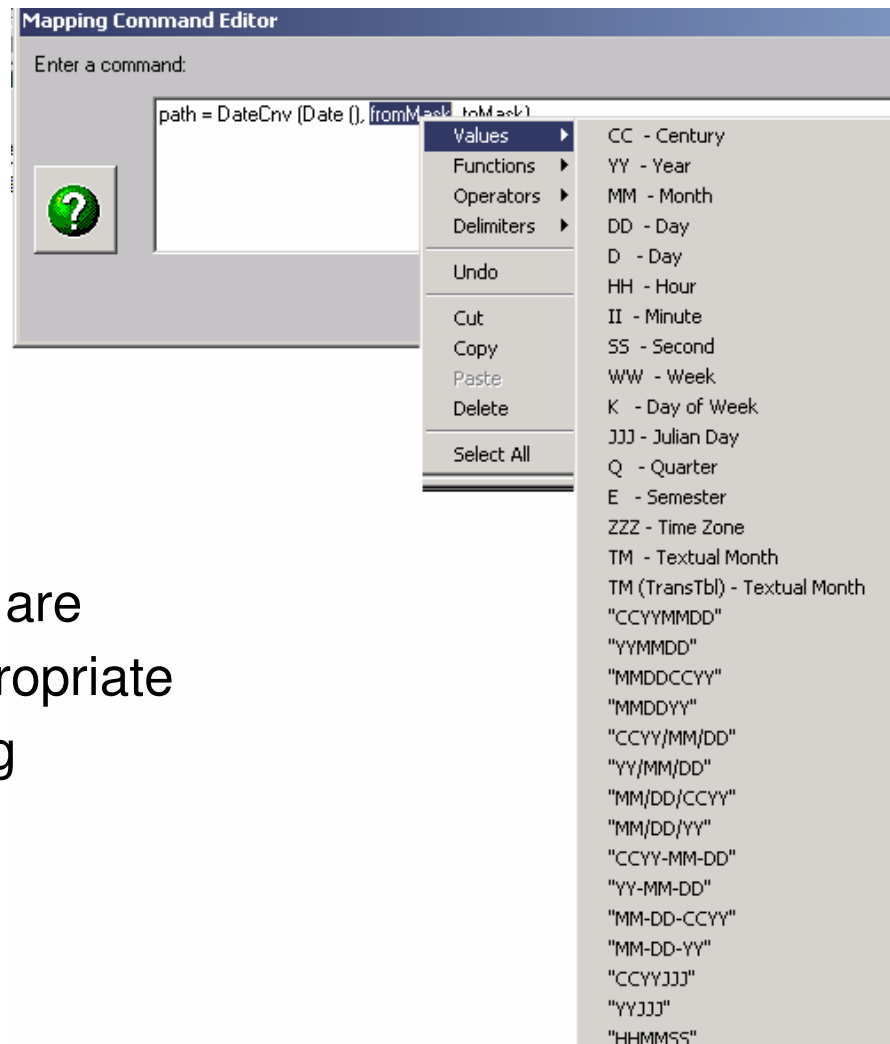


DT mapping functions

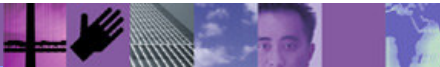
- Char
- Concat
- Created
- Date
- DateCnv
- Exit
- Find
- Found
- GetProperty
- HexEncode
- HexDecode
- IsEmpty
- Left
- Length
- Lower
- Number
- NumFormat
- Occurrence
- Overlay
- Right
- Round
- StrComp
- StrCompl
- StrCompN
- StrCompNI
- SubString
- Time
- Translate
- TrimLeft
- TrimRight
- Truncate
- Upper
- Validate



Right click for values

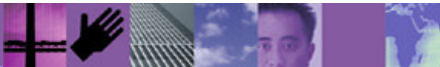


- Right click options are positional and appropriate for command being executed



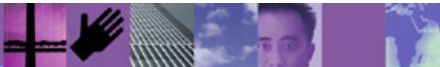
Expanded Conditional processing

- Additional functions available for conditional processing
 - StrComp, StrCompI, StrCompN, StrCompNI, Find, Found
 - Double Quotes used to compare character strings – “strings”
 - Single Quotes used to compare numeric fields – ‘123’



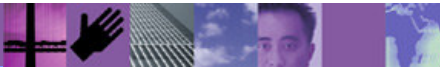
Simplified HL mapping

- Simplified HL mapping support
- Add Peer
- Add Child
- Automatically fills the remaining fields based on how you identified the parent/child relationships.



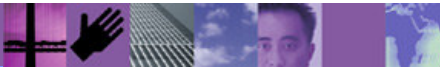
Complex Qualification Mapping Support

- Qualified mapping can be done by Occurrence, Value, or Expressions
- Combinations of these can also be used to achieve the desired results.



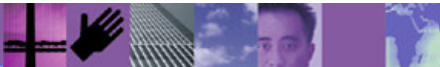
Integration

- Integration with WMQI using MCD profile to load MQ headers
- All processing and profiles handled from client
- Direct integration with TPI/Cyclone/WebSphere Business Integration Connect using EDICYCL network to drop data on a queue and passed to communications



Common Database

- WDI v3.2 uses a common database for WDI Server and WDI Client
 - allowing for easier installation
 - Simplified support
 - Fewer areas involved in installation and support



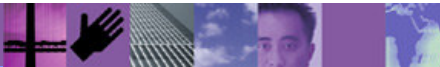
Better tracing capability

- Simplifies mapping problem definition
- Where **I** = the tracing level:
 - 0** All trace messages are ignored.
- Where **c** = component to be traced:
 - A** All nodes
 - D** Developer node
 - E** Enveloper node
 - M** Message broker
 - P** Parsers
 - R** Rules node
 - T** Transformation node
 - V** Validation node
 - 1** Normal tracing. Only the first 256 bytes of data in the buffer are written to the trace file.
 - 2** Extended tracing. The entire contents of the buffer is written to the trace file.



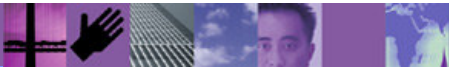
Any to Any mapping capability

- Data Transformation mapper allows:
 - DF to DF
 - EDI to EDI
 - XML to XML
 - CSV to CSV
 - Or any combination of the above



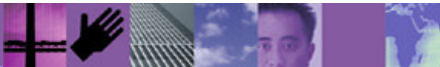
XML Support

- Allows the direct transformation of XML data
- Supports Schemas and DTDs
- No intermediate step to convert XML data



MQ Adapter

- Multi-threaded MQ adapter
 - Allows faster throughput during peak times
 - Minimizes DB contention



Summary

- In summary, WDI v3.2 offers:
 - Increased functionality
 - More mapping functions and features
 - Easier maintenance
 - Single pass any to any transformation
 - Easier problem resolution for mapping errors
 - Tighter integration
 - Expanded comments

