IBM Software Group

IBM WebSphere® Data Interchange V3.3

Multiple Map Execution

@business on demand.

© 2007 IBM Corporation

This presentation will describe multiple map execution features using Data Transformation maps.

Agenda

- Review MapSwitch, MapChain, MapCall

- Demonstrate how to setup multiple map execution

The presentation will review the MapSwitch, MapChain, and MapCall mapping commands for multiple map execution. Although the examples in this presentation are using WebSphere Data Interchange (WDI) Client Version 3.2, the mapping commands also apply to version 3.3.
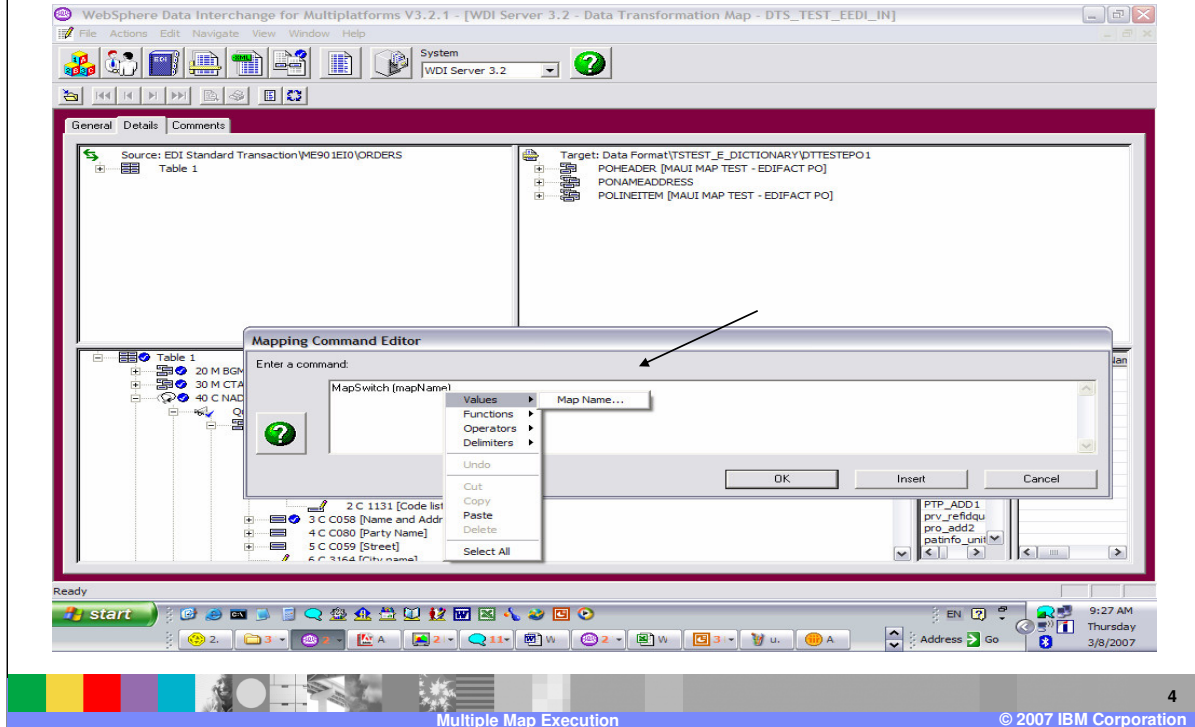
Multiple Map Execution

- MapSwitch = DIMAPSWITCH "value" command. For Receive maps the "value" is the "Application Sender" value on the Receive Usage of the map translation shoud switch or change to.
  - ▶ Use the MapSwitch command to indicate that the document needs to be translated by another map instead of the current map.
  - ▶ Syntax: MapSwitch (mapName)

- MapChain = DIMAPChain "value" command. For Receive the "value" is the "Application Sender" value on the Receive Usage of the map translation should switch or change to.
  - ▶ Use the MapChain command to indicate that the document needs to be translated by another map after the current translation has completed.
  - ▶ Syntax: MapChain (mapName)

3

The MapSwitch command indicates that the document needs to be translated by a different map instead of the current map. The command is similar and provides the same function as the Receive mapping command DIMAPSWITCH. With the MapSwitch command the current map execution is stopped, any output generated is cleared and the translation switches to the map identified. The MapChain command indicates that the document needs to be translated by another map after the current translation has completed. The command is similar and provides the same function as the Receive mapping command DIMAPCHAIN.

Multiple Map Execution

With both MapSwitch and MapChain, the mapName argument is a character string that specifies the map name. You can specify a literal value or a more complex expression that evaluates to a character string.

Multiple Map Execution

The MapSwitch and MapChain commands will most likely be used within conditional processing logic. With this example if the Party Identification equal 123456 the MapSwitch command will be executed and the translation will move to the map NewMap. The map NewMap should have the same source document, but the target document could be different.

Multiple Map Execution

- MapCall = No Send/Receive command. Allows Execution of a sub-map and returns the results to the calling map.

- Use the MapCall command to indicate that a new map must be used to process the data within the current source element (for source-based maps) or the specified source element (for target-based maps).

- Syntax Source Based: MapCall (This, mapName, targetPath)

- Syntax Target Based: MapCall (sourcePath, mapName, targetPath)

6

The MapCall command indicates that a sub-map or new map must be used to process the data within the current source element for source based maps or the specified source element for target based map. The first argument is sourcePath and identifies a source element that is to be used as the root of the source tree for the imbedded map. If the *sourcepath* element does not exist in the input message, the MapCall command is not executed. For source-based maps, this value must be the keyword *This*, which indicates the current source element. For target-based maps, the *sourcepath* element must be within the current domain.

The mapName argument is a character string that specifies the map name. You can specify a literal value or a more complex expression that evaluates to a character string.

The targetPath argument identifies the target element where the output of the imbedded map goes. If this is a simple binary element, such as a BIN02, the output from the called map is serialized and placed into this element. Use of compound elements and other data types for the targetPath on the MapCall command is not supported. If this argument is the keyword *targetRoot*, the following logic is applied: If the target document on the sub-map is the same as the target document on the primary map, the sub-map output is inserted into the same message as the primary map output. Children of the root in the sub-map output become children of the root in the primary map output. If the target document on the sub-map is different than the target on the primary map, a separate message output is created.

Multiple Map Execution

- MapCall()

- When this command is encountered, a new copy of the translator is loaded. It receives the data from the source element to use as its input document. The element can be a simple element (for example, the BIN02 element in a BIN segment) or it can be a compound element (a sub tree or subset).

- The copied translator shares global variables with the parent translator. When the copied translator has completed translation, it terminates, and control is returned to the parent translator.

When the MapCall command is encountered, a new copy of the translator is loaded. It receives the data from the source element to use as its input document. The element can be a simple element (for example, the BIN02 element in a BIN segment) or it can be a compound element (a sub tree or subset).

The copied translator shares global variables with the parent translator. When the copied translator has completed translation, it terminates, and control is returned to the parent translator.

Section

Source Based
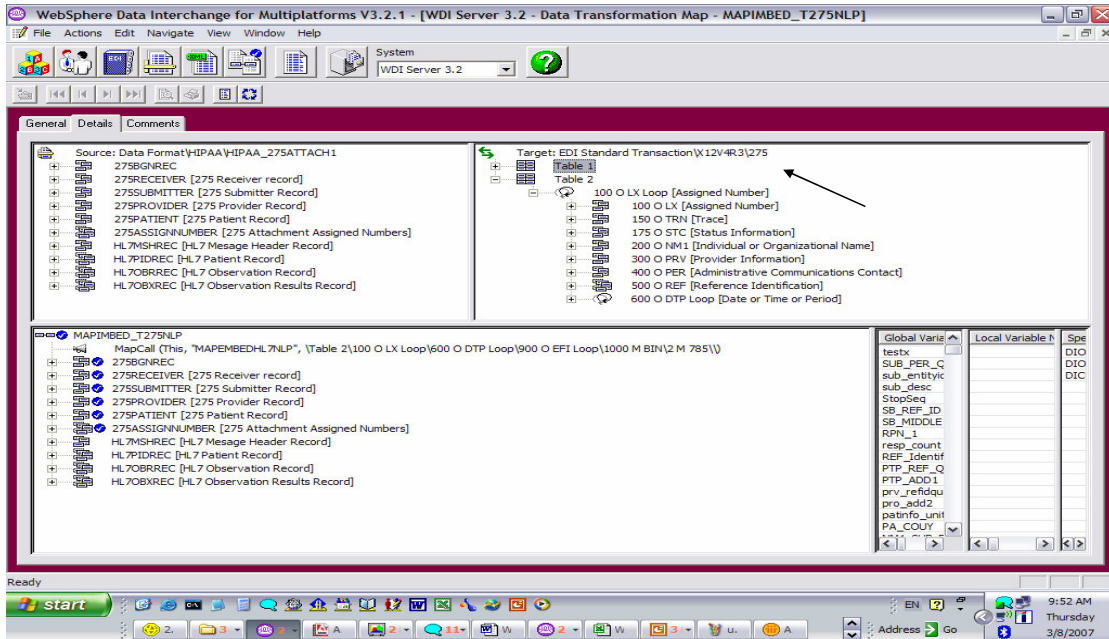
Multiple Map Execution

- ST*275*1002*004020X107~
- BGN*11*0001*19971015~
- NM1*PR*2*ABC INSURANCE COMPANY*****PI*05440~
- NM1*41*2*XYZ SERVICE*****46*A222222221~
- NM1*1P*1*FITCH*ROBERT*D**XX*KA6663~
- NM1*QC*1*JONES*PETER*M***HN*123456789A~
- REF*EJ*26463774~
- LX*1~
- TRN*2*987654~
- REF*CPT*44499~
- DTP*472*D8*19971003~
- DTP*368*D8*19971015~
- CAT*AE*HL~
- EFI*05~
- BIN*3231*MSH|^~\&|||||19980919131523||ORU^R01|
- A12349282|P|2.3|||NE|NE<cr>
- PID|||100928782^9^M11||JONES^PETER<cr>
- OBR||||11504-8^OPERATIVE NOTE^LN<cr>
- OBX||TX|11504-8||JONES, PETER~

This is an Electronic Data Interchange (EDI) document example of the X12 275 transaction. The 275 format is the same as any other document but contains a different EDI document embedded within the BIN segment beginning with the MSH segment. This requires a mapping execution to create the primary data in the 275 transaction and a different mapping execution to create the embedded EDI data.

Multiple Map Execution

This example is a source based map. The source document is application data and the target document is EDI dictionary X12V4R3 and document 275.

Multiple Map Execution

WebSphere Data Interchange for Multiplatforms V3.2.1 - [WDI Server 3.2 - Data Transformation Map - MAPIMBED_T275NLP]

File   Actions   Edit   Navigate   View   Window   Help

System
WDI Server 3.2

General   Details   Comments

Source: Data Format\HIPAA\HIPAA_275ATTACH1

275BGNREC
275RECEIVER [275 Receiver record]
275SUBMITTER [275 Submitter Record]
275PROVIDER [275 Provider Record]
275PATIENT [275 Patient Record]
275ASSIGNNUMBER [275 Attachment Assigned Numbers]
HL7MSHREC [HL7 Mesage Header Record]
HL7PIDREC [HL7 Patient Record]
HL7OBRREC [HL7 Observation Record]
HL7OBXREC [HL7 Observation Results Record]

150 O TRV [Trace]
175 O STC [Status Information]
200 O NM1 [Individual or Organizational Name]
300 O PRV [Provider Information]
400 O PER [Administrative Communications Contact]
500 O REF [Reference Identification]
600 O DTP Loop [Date or Time or Period]
  600 O DTP [Date or Time or Period]
  700 O CAT [Category of Patient Information Service]
  800 O PID [Product/Item Description]
  900 O EFI Loop [Electronic Format Identification]
    900 O EFI [Electronic Format Identification]
    1000 M BIN [Binary Data]
      1 M 784 [Length of Binary Data]
      2 M 785 [Binary Data]

MAPIMBED_T275NLP
  MapCall (This, "MAPEMBEDHL7NLP", \Table 2\100 O LX Loop\600 O DTP Loop\900 O EFI Loop\1000 M BIN\2 M 785\\)
  275BGNREC
  275RECEIVER [275 Receiver record]
  275SUBMITTER [275 Submitter Record]
  275PROVIDER [275 Provider Record]
  275PATIENT [275 Patient Record]
  275ASSIGNNUMBER [275 Attachment Assigned Numbers]
  HL7MSHREC [HL7 Mesage Header Record]
  HL7PIDREC [HL7 Patient Record]
  HL7OBRREC [HL7 Observation Record]
  HL7OBXREC [HL7 Observation Results Record]

Global Varia
testx
SUB_PER_Q
sub_entityic
sub_desc
StopSeq
SB_REF_ID
SB_MIDDLE
RPN_1
resp_count
REF_Identif
PTP_REF_Q
PTP_ADD1
prv_refidqu
pro_add2
patinfo_unit

Special Variable Nam
DIOutType
DIOutFile
DICUserData

Ready

start   EN   9:41 AM   Thursday   3/8/2007

Address   Go

11

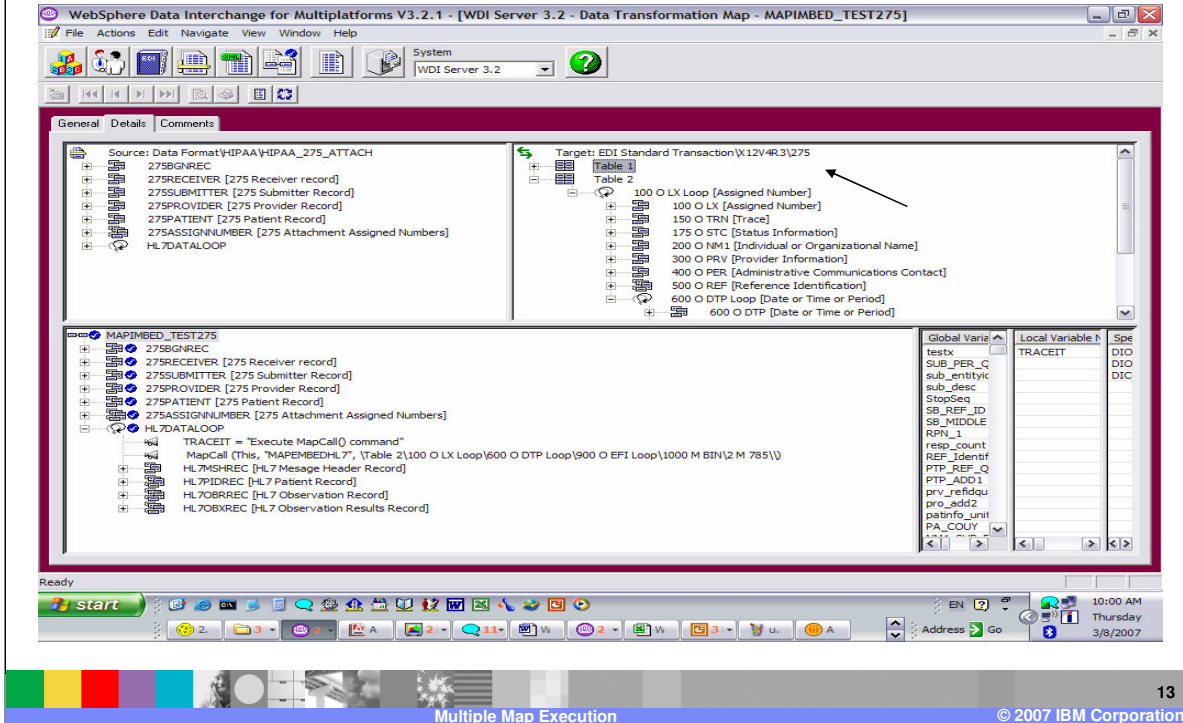Multiple Map Execution   © 2007 IBM Corporation

The MapCall command will pass the source root element to the called map "MAPEMBEDHL7NLP" and the results will be mapped to the BIN02 element. The keyword "This" identifies the current element as the source element. Since the command is within the source root element the source root element will be used.

Multiple Map Execution

Multiple Map Execution

© 2007 IBM Corporation
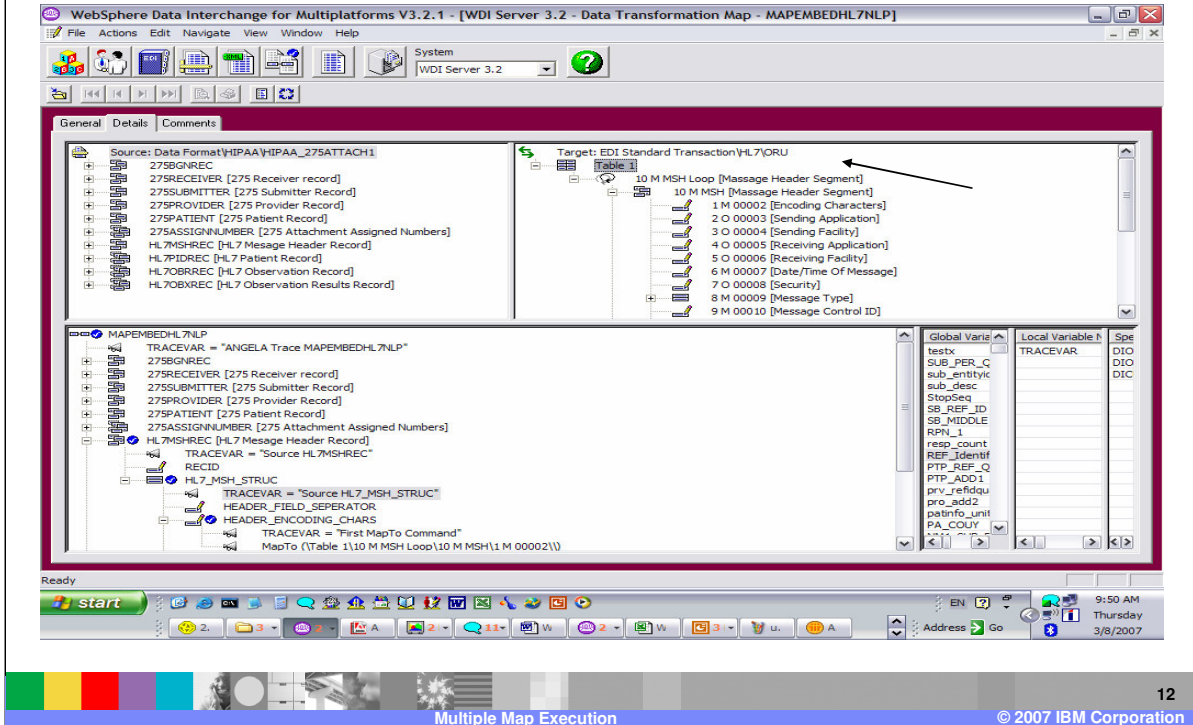
This is the called map "MAPEMBEDHL7NLP". The Source document is the same as the calling or primary map and the Target document is EDI dictionary HL and document ORU which is a different target document.

Multiple Map Execution

This example is a source based map. The source document is application data and the target document is EDI dictionary X12V4R3 and document 275.
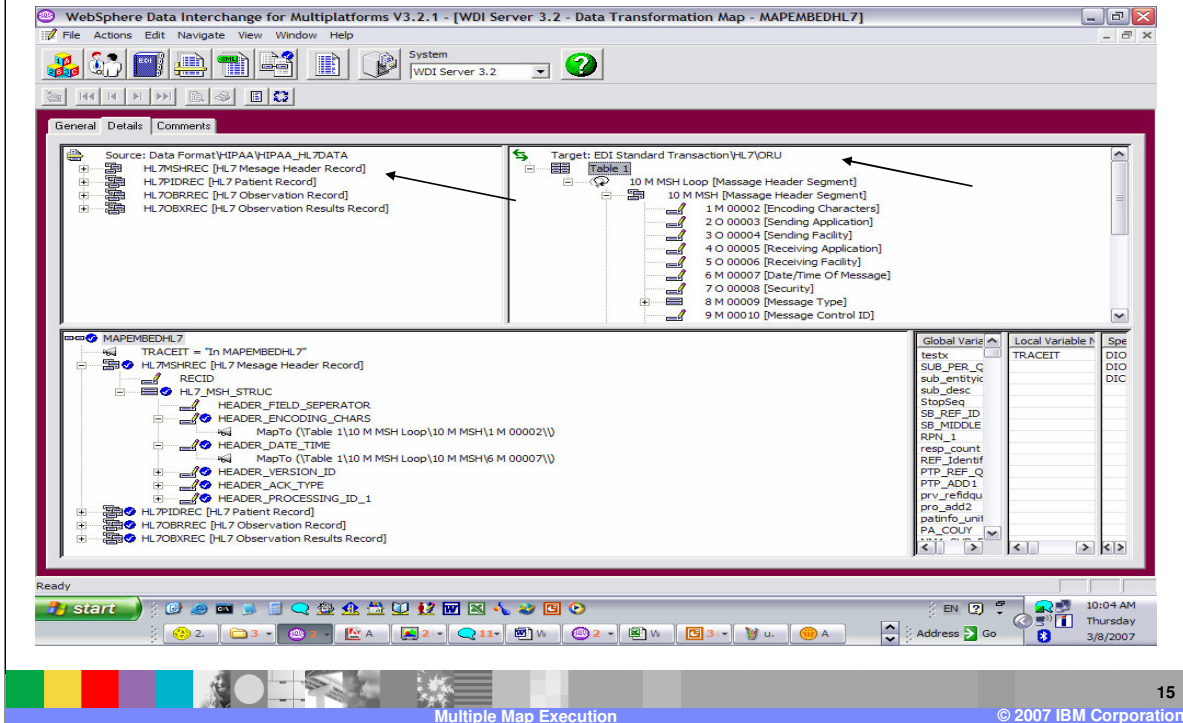
Multiple Map Execution

The MapCall command will pass the HLDATALOOP element to the called map "MAPEMBEDHL7" and the results will be mapped to the BIN02 element. The keyword "This" identifies the current element as the source element. Since the command is within the HLDATALOOP element this element will be used. The HL7MSHREC is the first element within the HLDATALOOP.

Multiple Map Execution

This is the called map "MAPEMBEDHL7". The Target document is EDI dictionary HL and document ORU which is a different target document than the calling map. The Source document is a sub-set of the original source document in the primary map. The HL7MSHREC is the first element within the HLDATALOOP. The results of the mapping in this map will be returned to the primary map.
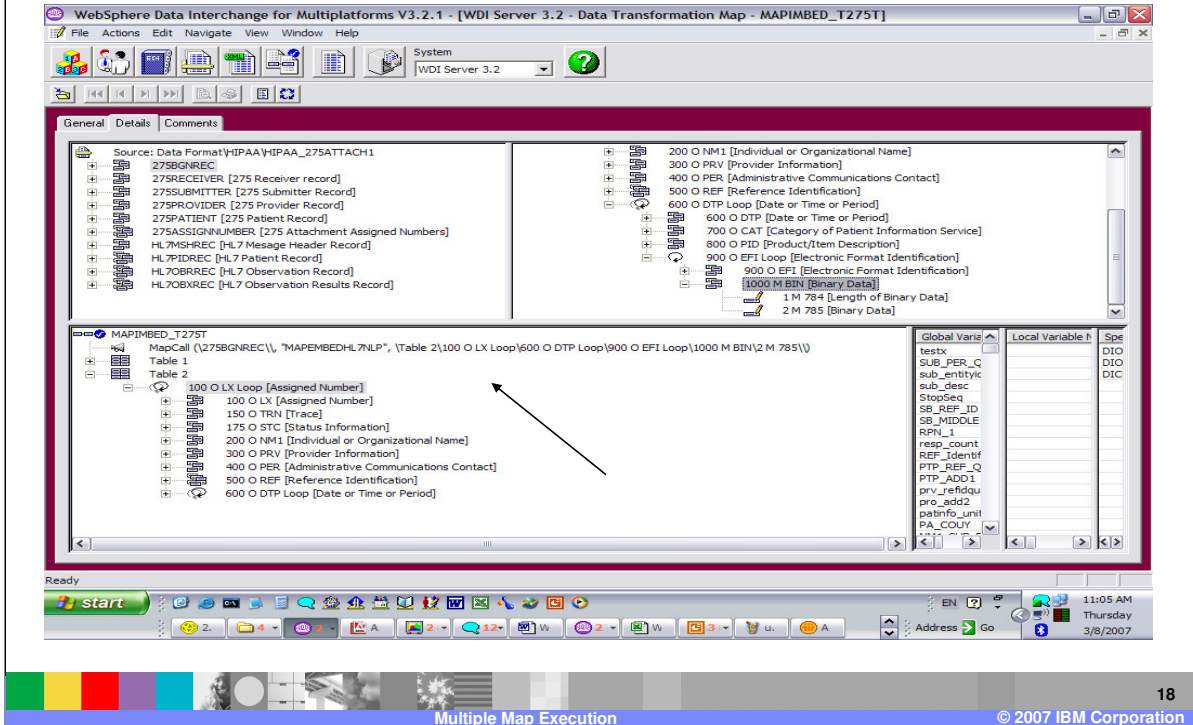
Section

Target Based

Multiple Map Execution

This example is a target based map. The source document is application data and the target document is EDI dictionary X12V4R3 and document 275.
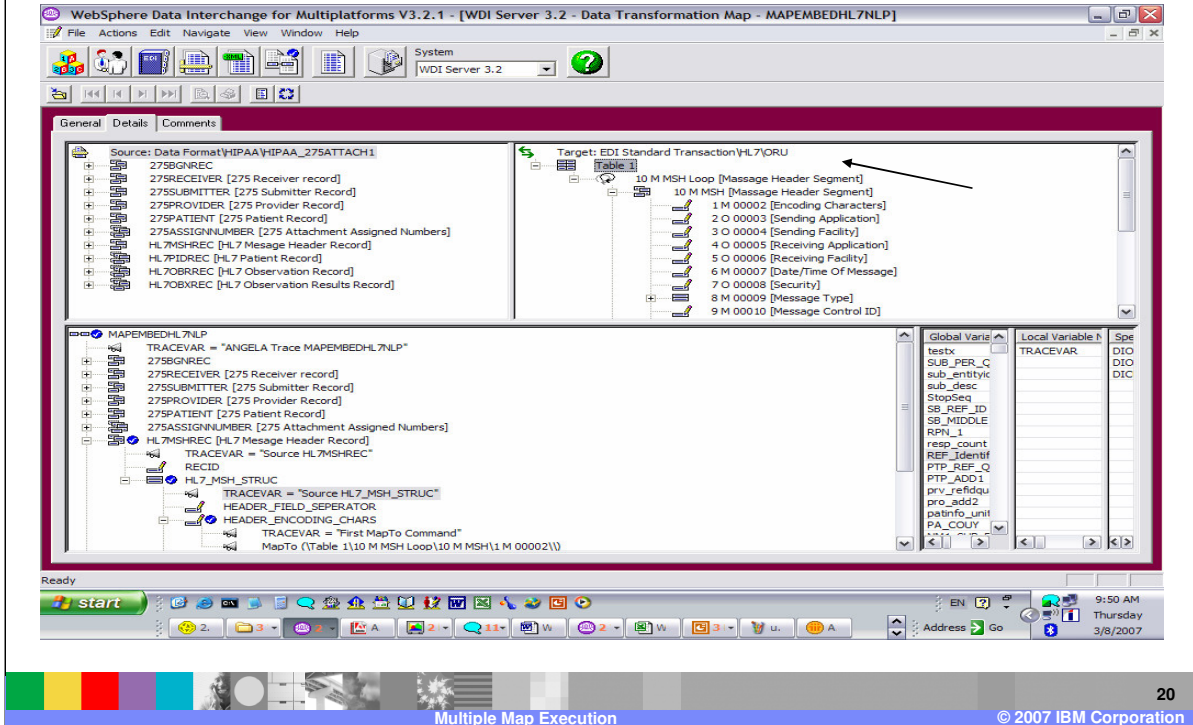
Multiple Map Execution

The MapCall command will pass the source element 275BEGREC to the called map "MAPEMBEDHL7NLP" and the results will be mapped to the BIN02 element.

Multiple Map Execution

The *sourcepath* element must be within the current domain. With this example there is a multiple occurrence qualification under the DTP loop using the ForEach command. The source element is not within the current domain and results in a control string compile error.

Multiple Map Execution

This is the called map "MAPEMBEDHL7NLP". The Source document is the same as the calling map and the Target document is EDI dictionary HL and document ORU which is a different target document.

Multiple Map Execution

This example is a target based map. The source document is application data and the target document is EDI dictionary X12V4R3 and document 275.

Multiple Map Execution

The MapCall command will pass the HLDATALOOP element to the called map "MAPEMBEDHL7" and the results will be mapped to the BIN02 element. The HL7MSHREC is the first element within the HLDATALOOP.

Multiple Map Execution
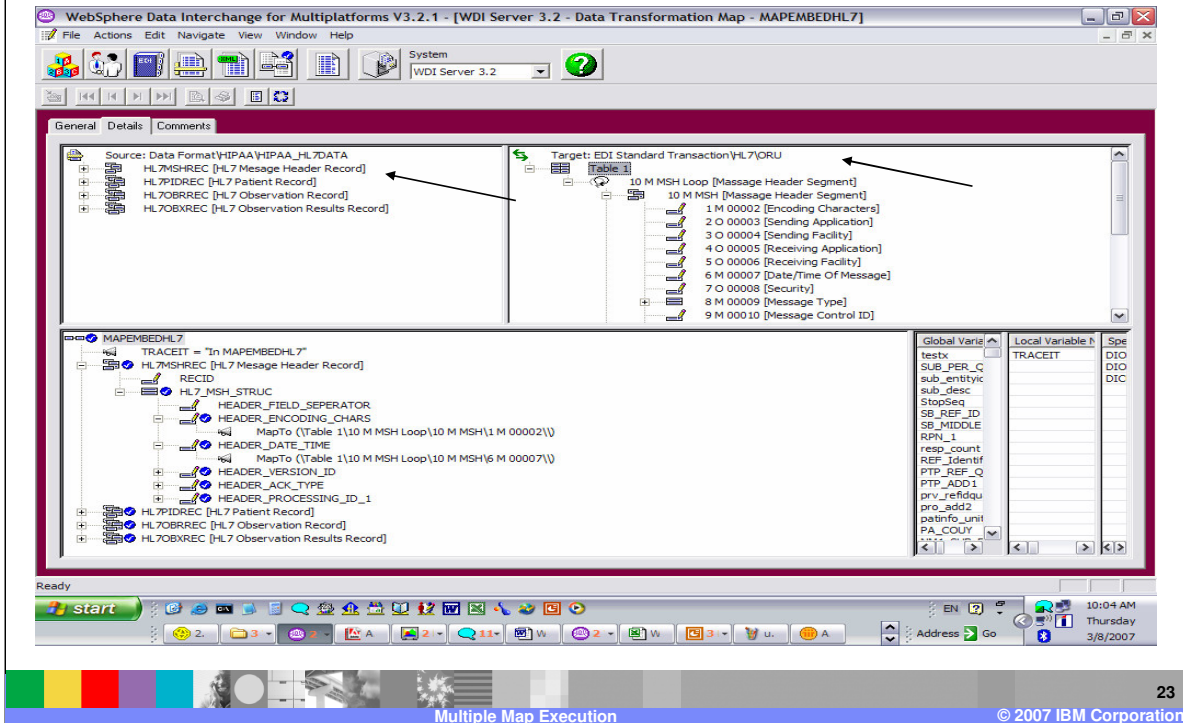
This is the called map "MAPEMBEDHL7". The Target document is EDI dictionary HL and document ORU which is a different target document than the calling map. The Source document is a sub-set of the original source document. The HL7MSHREC is the first element within the HLDATALOOP.

Reference

- More information can be found in the WDI V3.3 Mapping Guide.

More information can be found in the WebSphere Data Interchange Version 3.3 Mapping Guide.

IBM Software Group

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	WMQ	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.