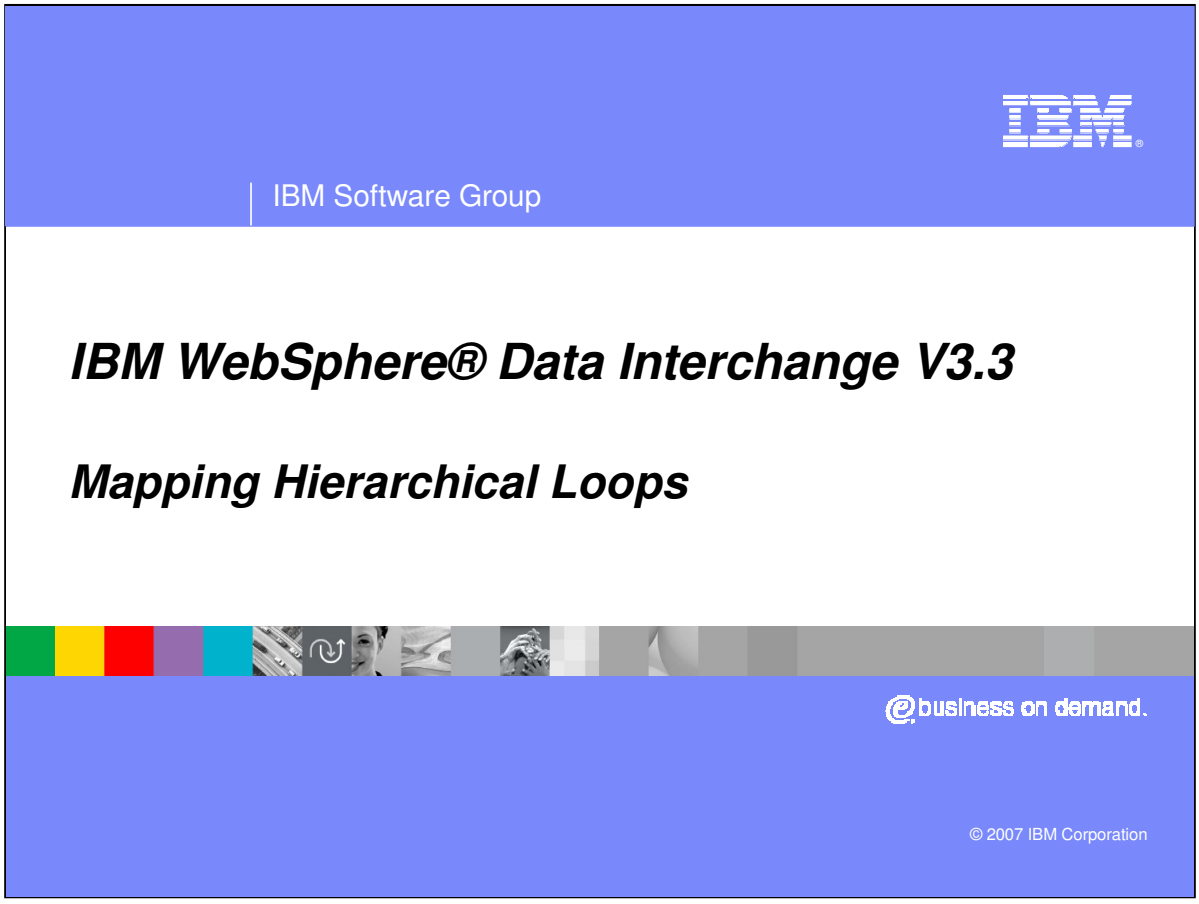


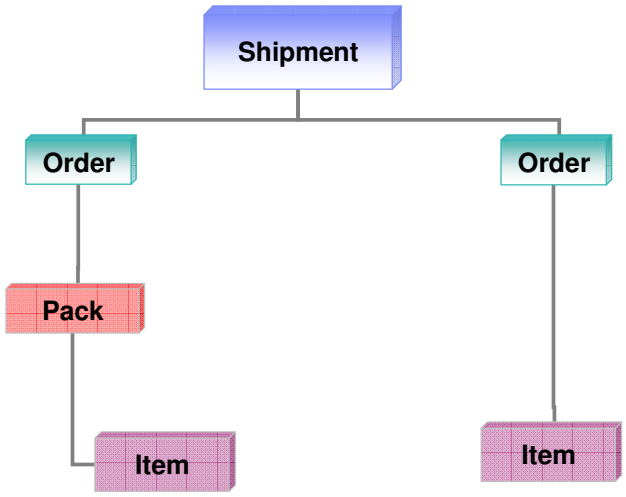
This presentation will illustrate how to map Hierarchical Loops in Send and Receive maps.

Agenda

- What is special HL Loop mapping?
- Selecting special HL Loop mapping.
- Qualification.
- Add parent child relationships.
- Summary and references

The presentation will describe Hierarchical Level (HL) Loops and describe how to use the WebSphere Data Interchange (WDI) special Hierarchical Loop Mapping commands.

Mapping Hierarchical Loops



This is an example of a hierarchy. The Shipment contains orders, orders contain packs or items, and packs contain items.

Mapping Hierarchical Loops

- What is special HL Loop Mapping?
  - The EDI HL segment contains values that identify each level of the hierarchy and the parent child relationships.
  - Specifically developed for Send (EDI target) but can be used for Receive (EDI source).
  - Should be used when your source data does not contain the values to be mapped to the HL segment.
  - WDI special HL Loop Mapping defines the hierarchy which allows the translation process to generate the HL segment values.

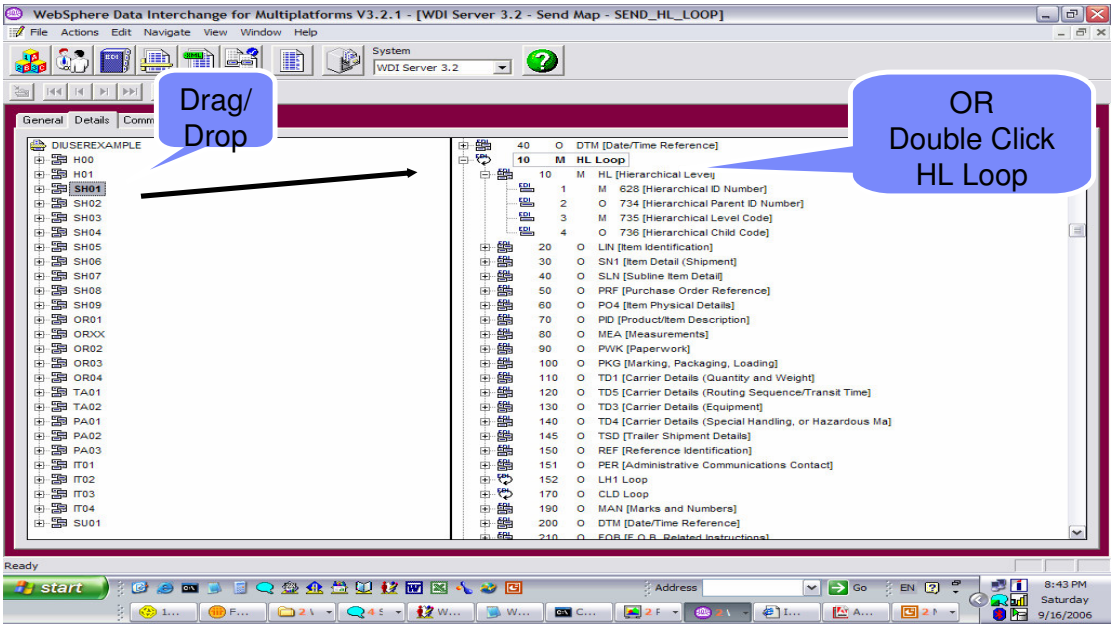
WDI provides special mapping functions to map the Electronic Data Interchange (EDI) standard HL segment. This special mapping is specifically for an EDI target document when the source document does not contain the values that define the hierarchy in the HL segment.

Mapping Hierarchical Loops

- Send and Receive mapping:
  - Allows you to specify unique mapping instructions for each identifiable group of structures in a hierarchical loop.
  - Can handle 16 levels of nesting within the HL loop structure.
- If special HL Loop mapping has been selected you must delete all mapping on the HL Loop to remove special HL Loop mapping.

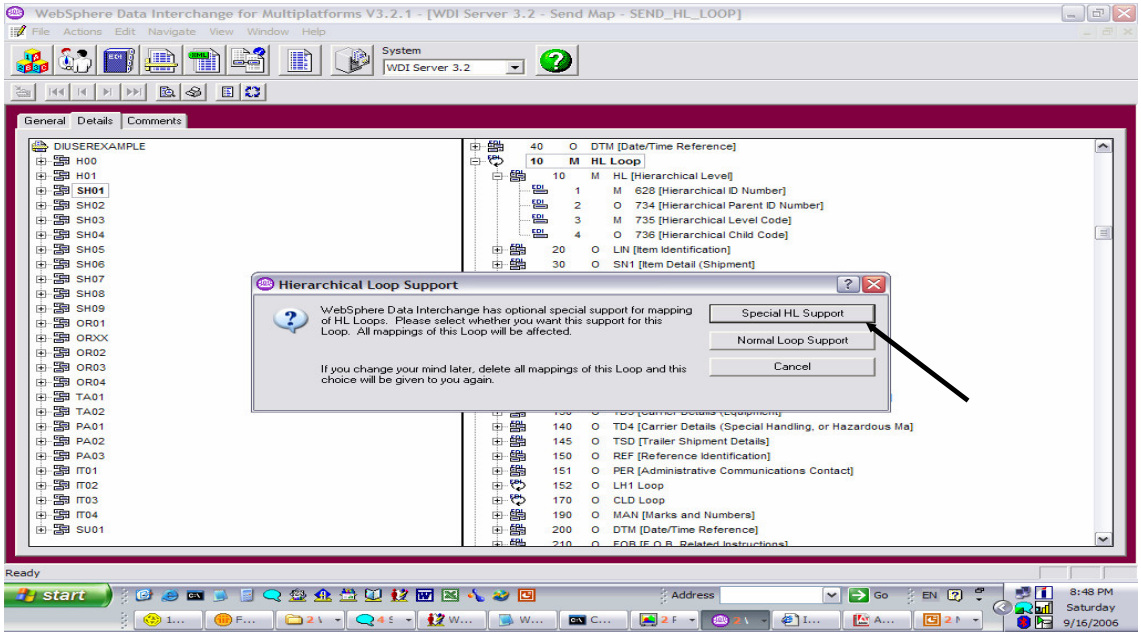
Send and Receive mapping allows you to specify unique mapping instructions for each identifiable group of structures in a hierarchical loop. The translation process can handle 16 levels of nesting within the HL loop structure. If special HL Loop mapping has been selected you must delete all mapping on the HL Loop to remove special HL Loop mapping.

Mapping Hierarchical Loops



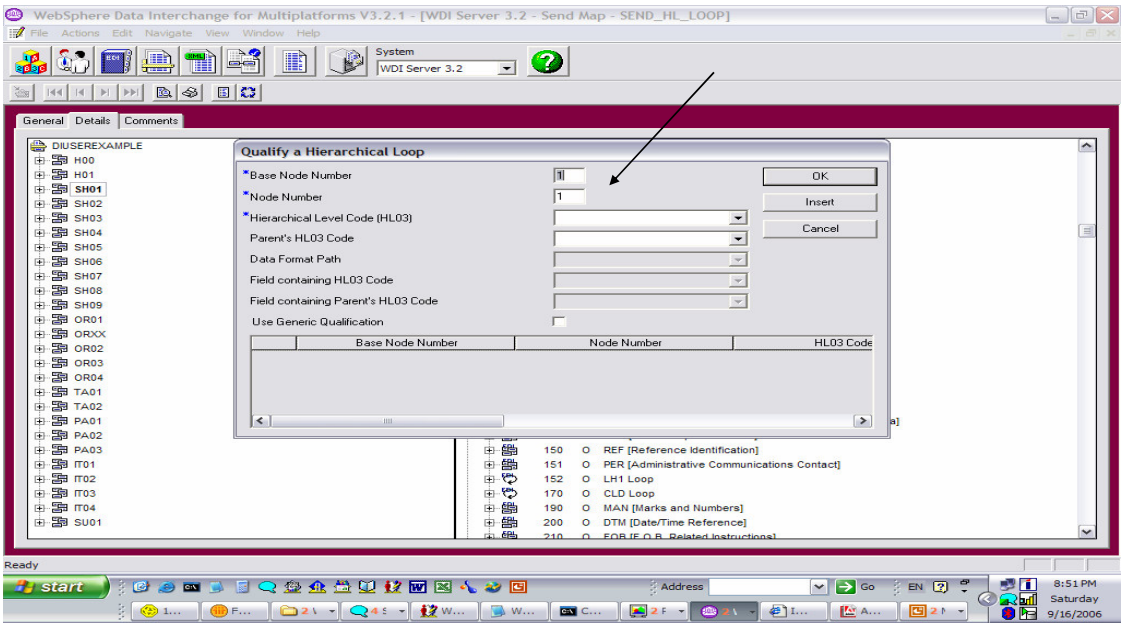
This is an example of a Send map which has EDI as the target document. To select special HL Loop mapping either use the drag and drop from the source window on the left to the target window on the right or double click the HL Loop in the right window.

Mapping Hierarchical Loops



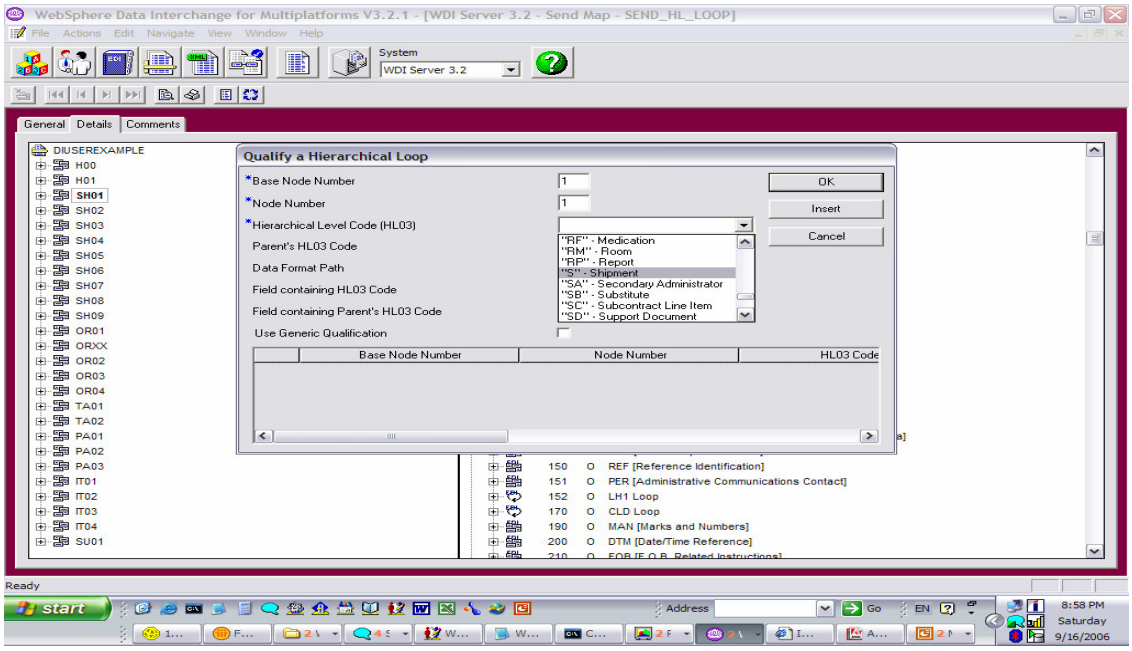
A window will display with a selection for Special HL Support.

Mapping Hierarchical Loops



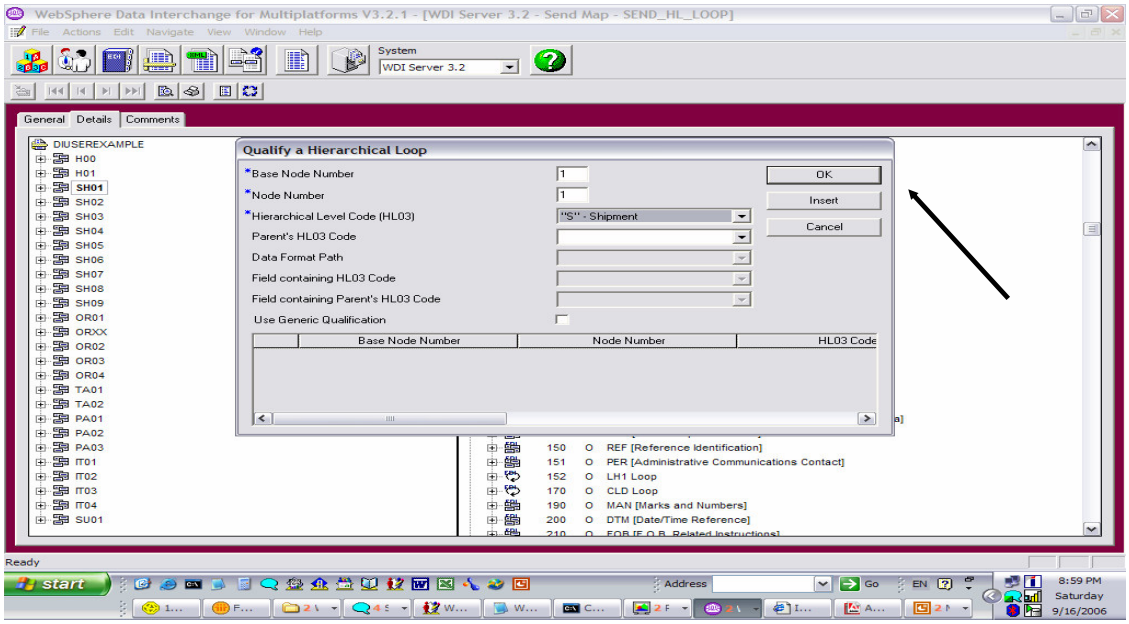
To begin, you identify the base level. This is done by specifying the Base Node Number and Node Number.

Mapping Hierarchical Loops



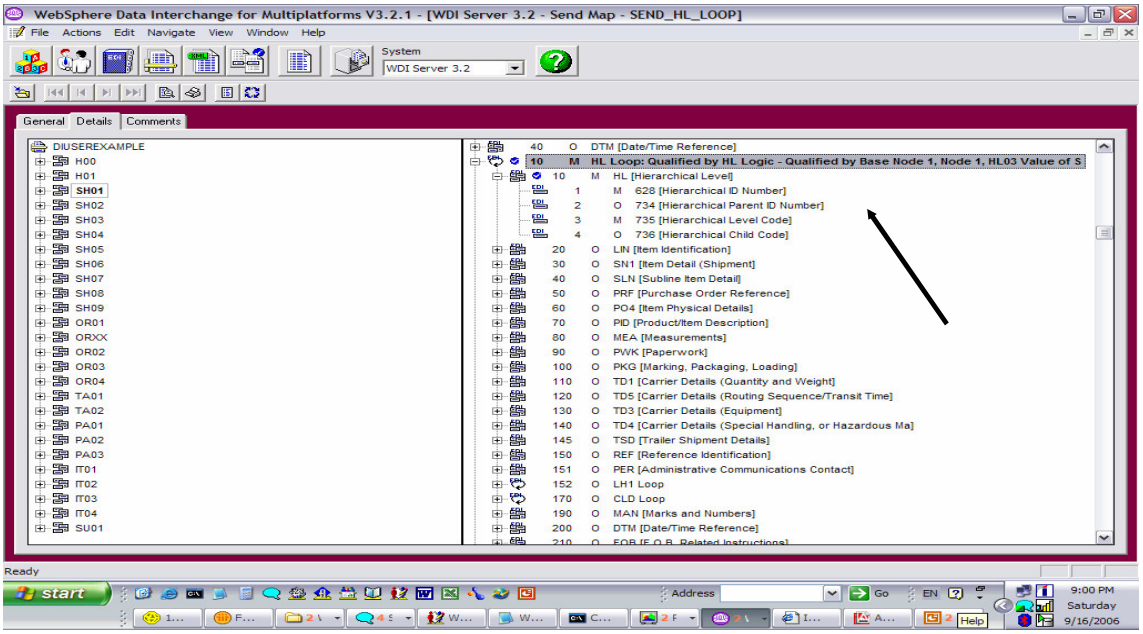
Next you identify the Hierarchical Level Code. The values in the drop down list come from the EDI code list for element 735.

Mapping Hierarchical Loops



Select the hierarchical level code from the list and press OK. Since this is the base level there is no parent HL Loop to define.

Mapping Hierarchical Loops



This results in an HL Loop Qualification.

Mapping Hierarchical Loops

- The HL segment is the first segment of the HL loop
- HL*ID Number*Parent ID Number*Level Code*Child Code!
  - ID Number Sequence number for HL loop
  - Parent ID Number Sequence number of parent HL loop
  - Level Code A code used to identify the instance of the HL loop
  - Child Code 0 or 1 indicates this instance of the HL loop has children

The HL segment is the fist segment of the HL loop. The ID Number is a sequence number, the parent ID number is the sequence number of the parent HL segment or loop, the level code is a code used to identify this instance of the HL loop, and the child code indicates if this instance has children.

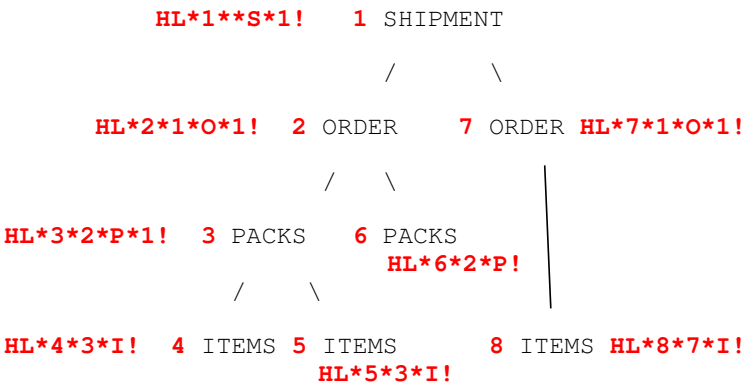
Mapping Hierarchical Loops

- Hierarchical Rules
  - Define hierarchical paths.
    - Number the nodes top to bottom and left to right starting with the left most node.
  - Identify base nodes.
    - Node 1 Shipment
  - Identify levels.
    - Shipment = S, Order = O, Packs = P, Items = I
  - Identify relationships
    - Is a Shipment required for an Order?

Here are some rules for defining a hierarchy before you begin mapping. You want to define the hierarchical paths, identify base nodes, and identify levels. Determine the relationships between the levels for example is a shipment level required for the order level or can the order contain an item level without a pack level.

Mapping Hierarchical Loops

- HL segment
- HL*ID Number*Parent ID Number*Level Code*Child Code!



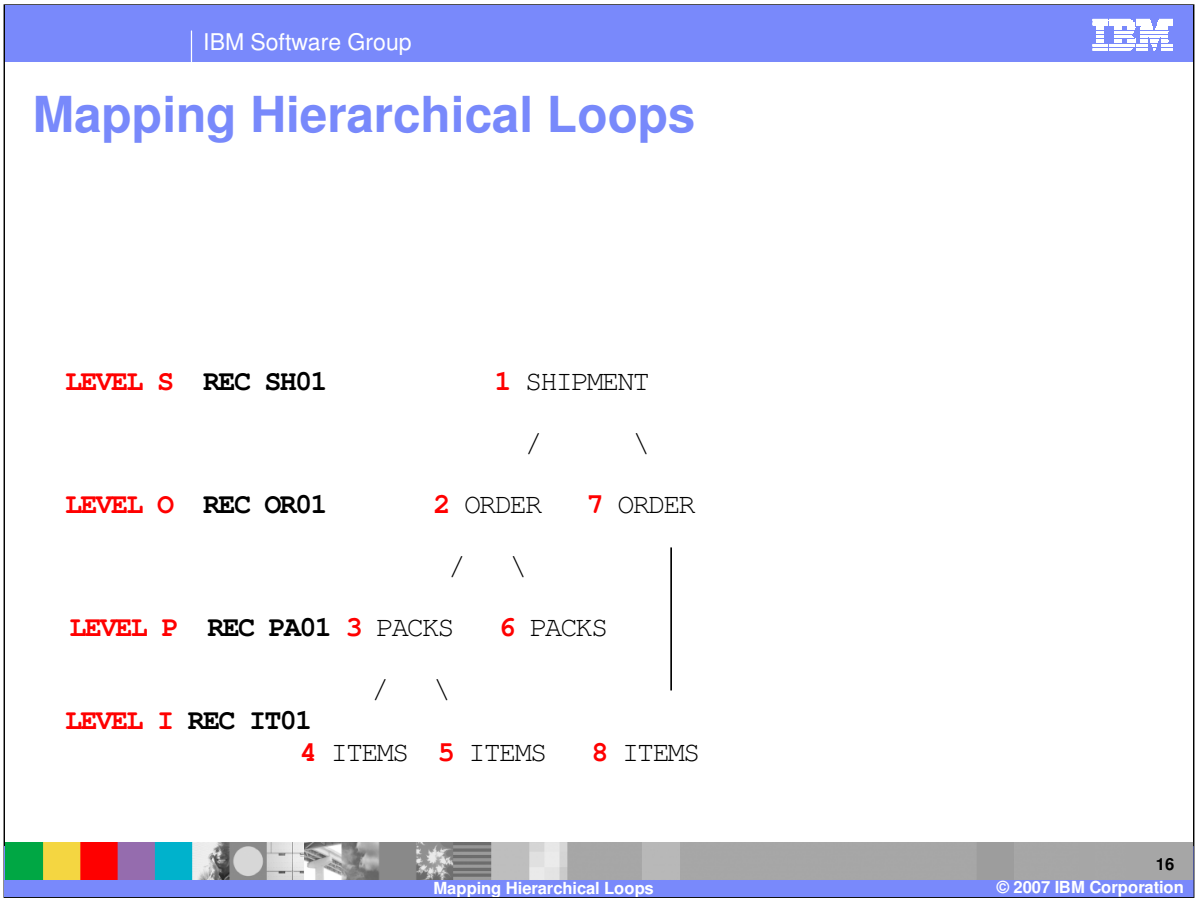
Number the nodes top to bottom and left to right starting with the left most node.

Mapping Hierarchical Loops

- Sample application data

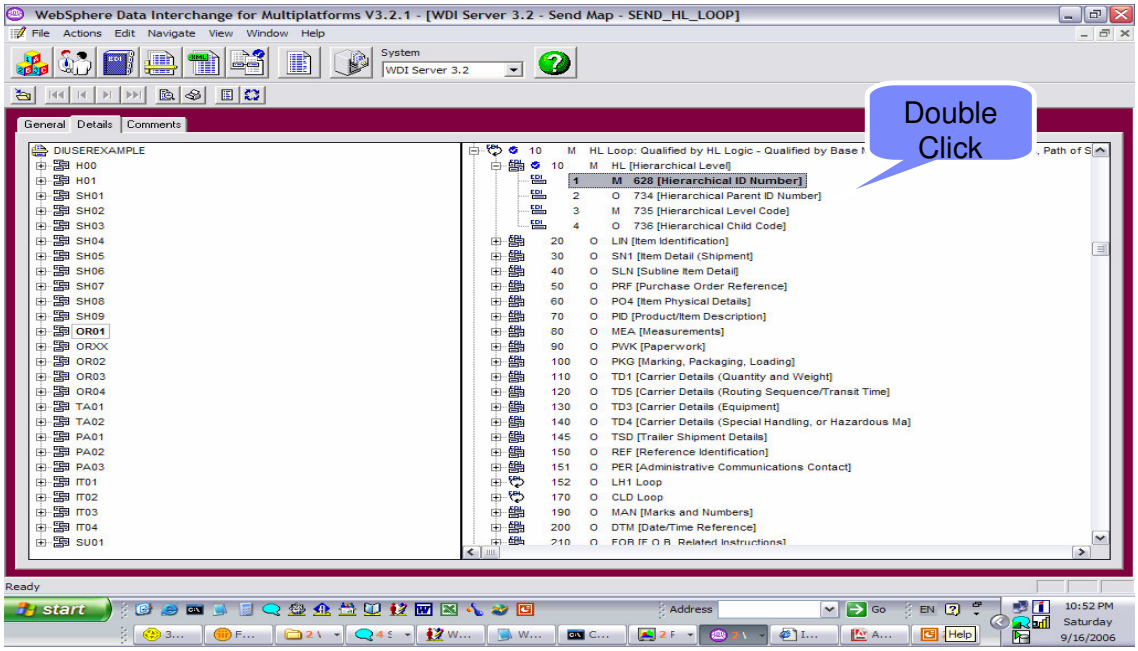
```
H00 DIUTESTPTR     01003020VICS  940830170700000170900062921
H01 00131313131313                  9407141416  0001
SH011                        S
SH02CTN250000072
OR012           1            O
OR02887378
OR03B                     C FOREWAY
OR04BYHLLOOP,INC.                           9 DUNSNUMBR0897
PA013 2            P
PA0200000600000000          000000000054000LB000000000000000
IT014            3            I
IT02                                                UP022222640304
IT03            0000000006EA00000000000000000006EA
```


This is an example of application data that will be mapped to the HL Loop.



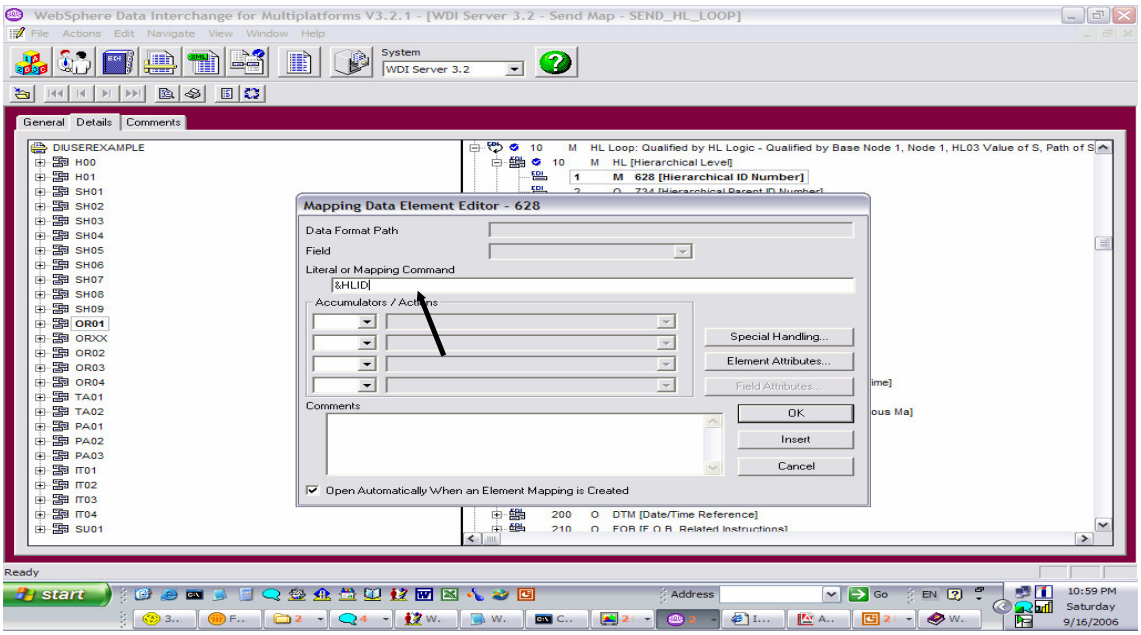
If you use the rules and identify the levels and the relationships, the mapping process will be easier. The hierarchy defined in this example contains 2 possible hierarchies for the Order level. Orders can have Packs and Items or Items. This will require HL Loop mapping for Items when Packs is the parent and Items when Order is the parent. Therefore Items is a child of Packs. Items is a child of Order and also a sibling of Packs.

Mapping Hierarchical Loops



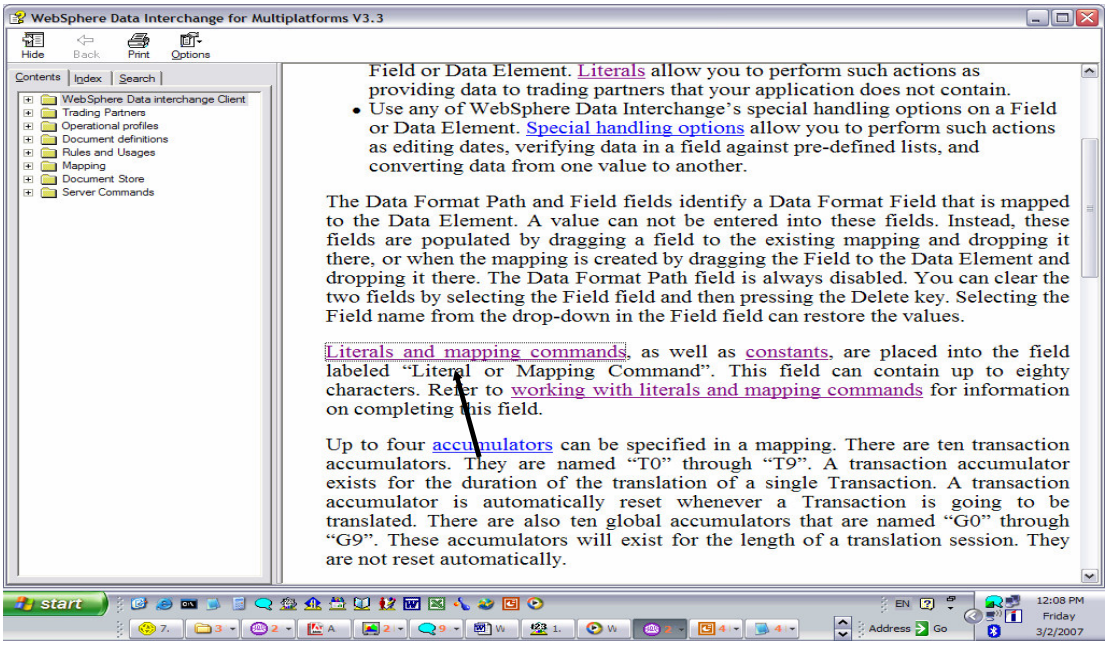
To map the elements that define the hierarchy, double click on each element.

Mapping Hierarchical Loops



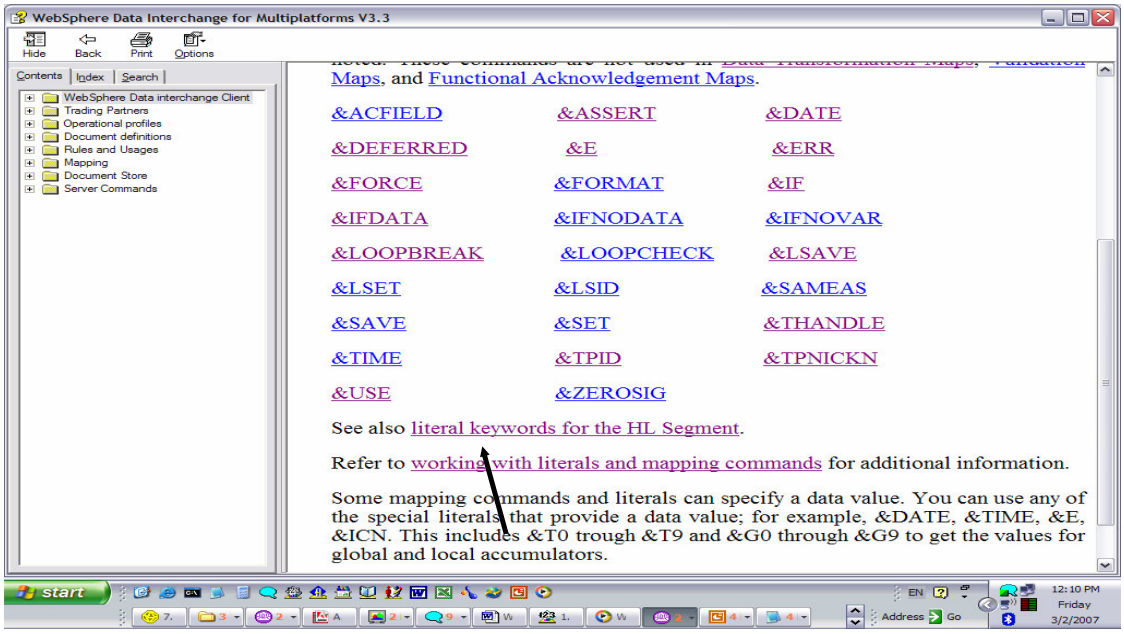
WDI supplies special mapping commands to automatically map the elements in the HL segment. You can press the F1 Key on your keyboard to view the client help.

Mapping Hierarchical Loops



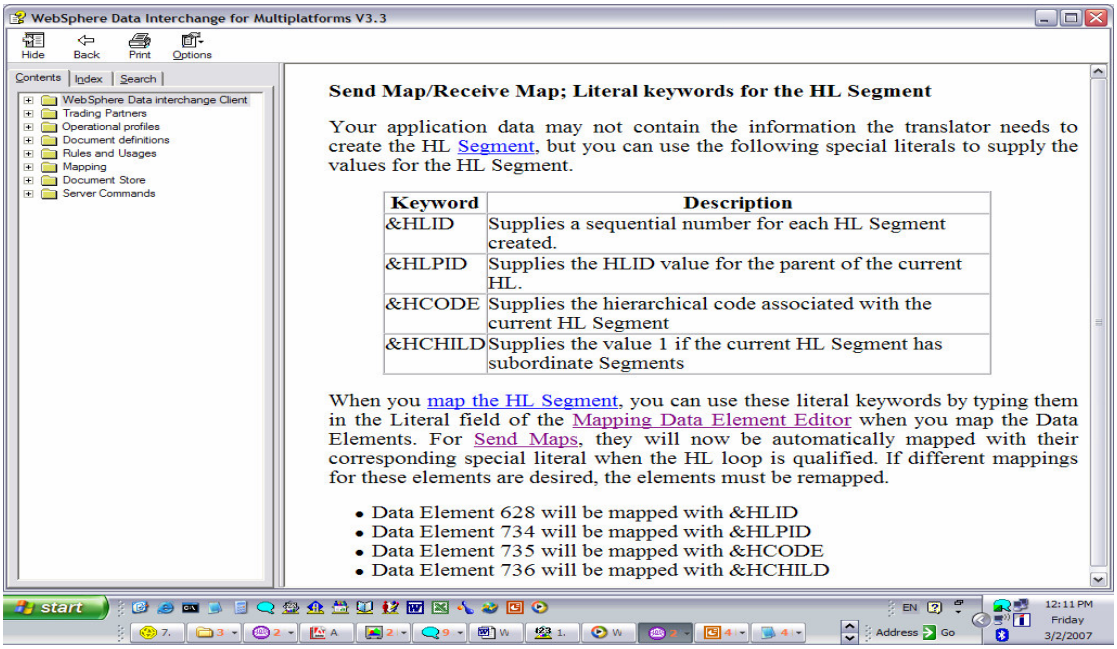
This example will show the WDI version 3.3 help. This help is also available using WDI version 3.2. Select Literals and mapping commands.

Mapping Hierarchical Loops



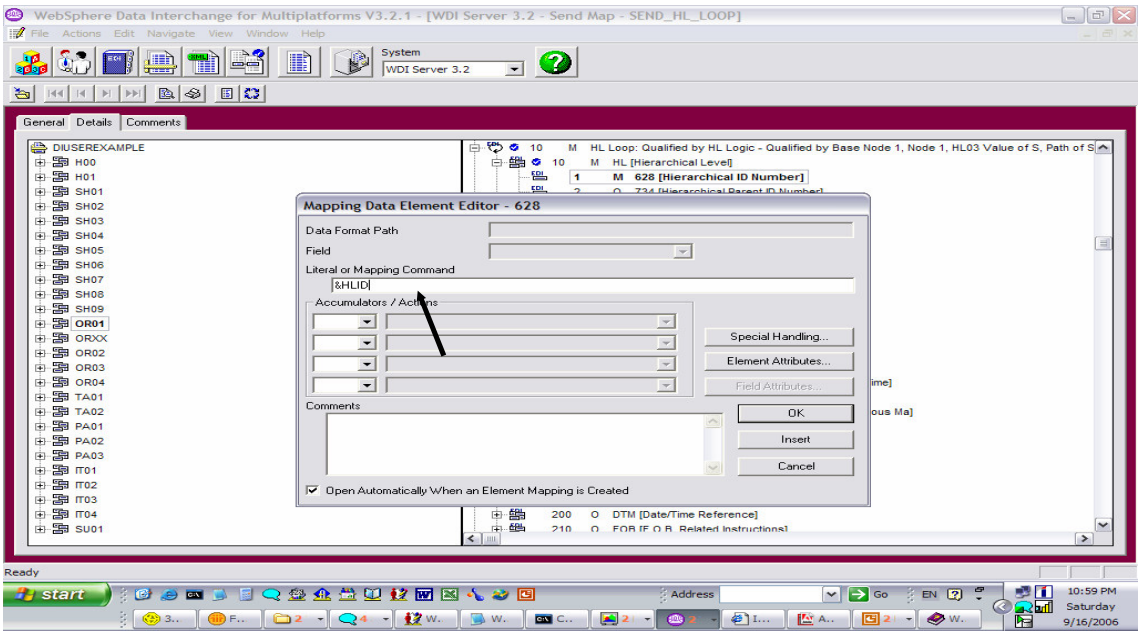
Select Literals keywords for the HL Segment.

Mapping Hierarchical Loops



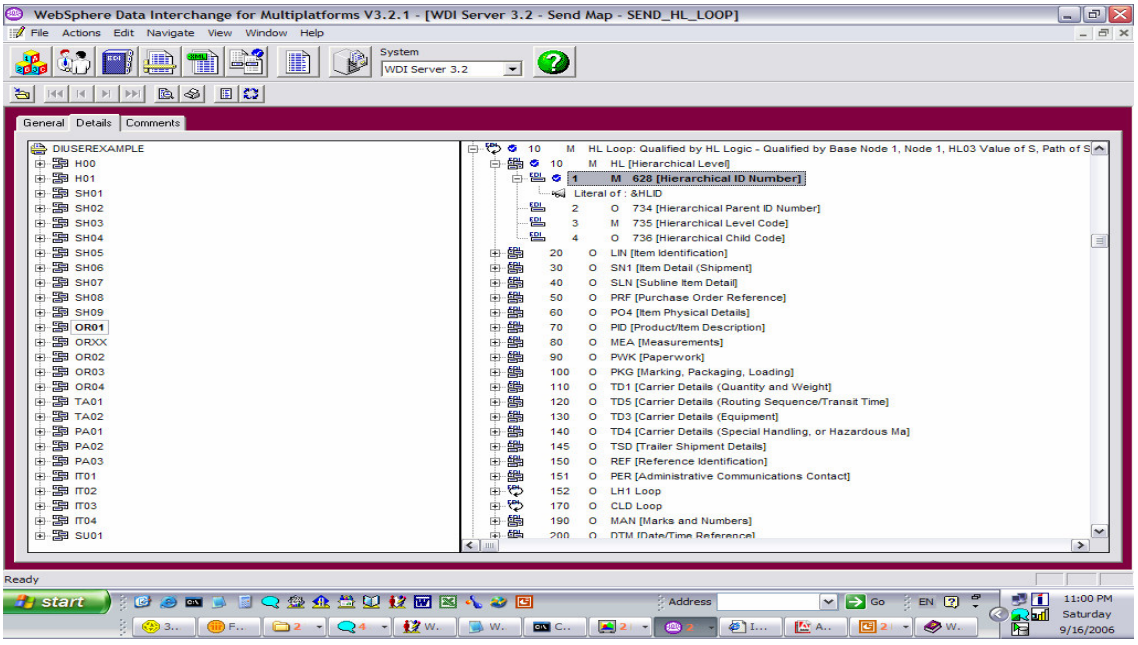
The &HLID keyword signals the translation to supply a sequential number for each HL segment created. &HLPID signals the translation to supply the sequence number of the parent for the current HL segment. &HCODE supplies the hierarchical code for the current HL segment. And &HCHILD supplies the value 1 if the current HL segment has children.

Mapping Hierarchical Loops



Enter the &HLID keyword to signal the translation process to generate this value.

Mapping Hierarchical Loops



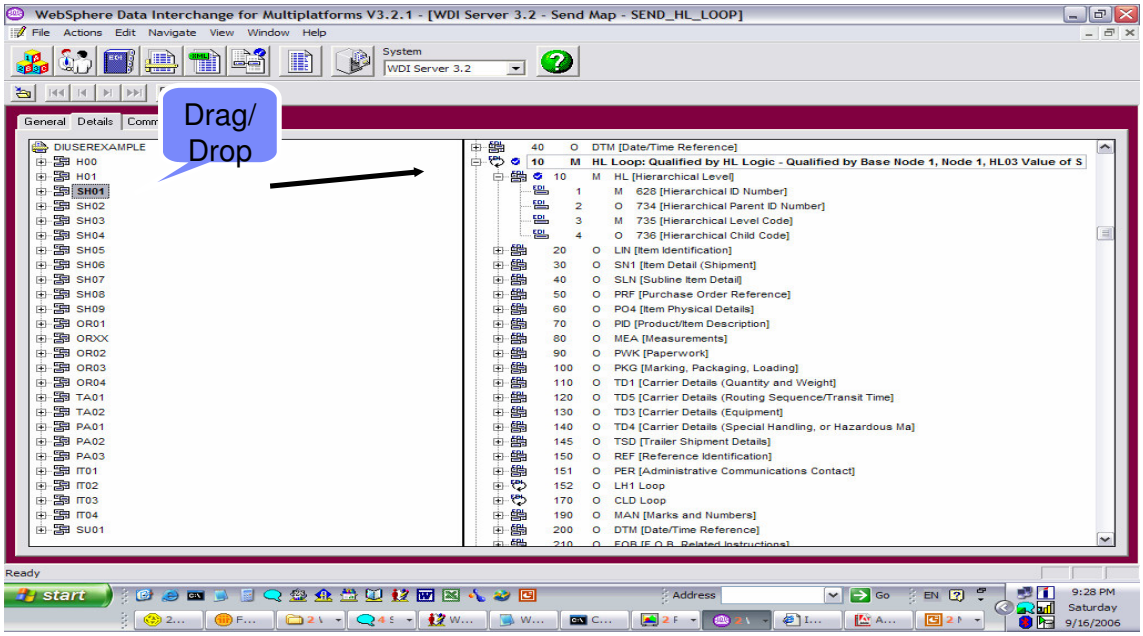
Here is the results of mapping the &HLID keyword. Continue mapping the other elements and supply the appropriate keywords for automatic generation of these values.

Mapping Hierarchical Loops

- Qualification HL Loop Send Mapping
  - Multiple Occurrence
  - Single Occurrence
  - No Value Qualification

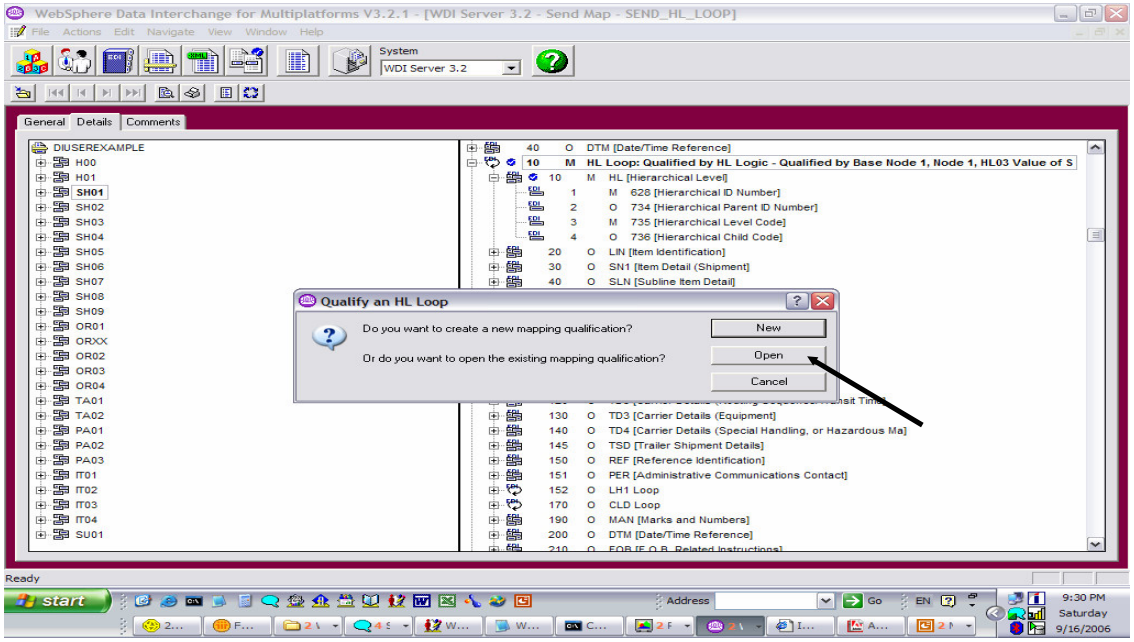
You can add qualification for each HL mapping.

Mapping Hierarchical Loops



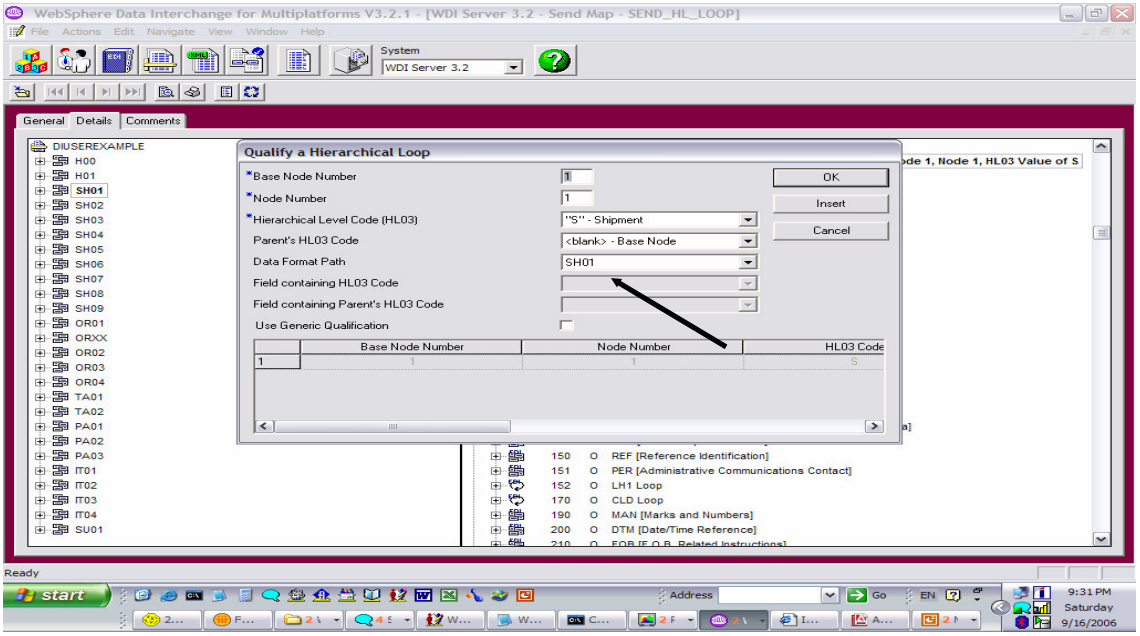
You can use drag and drop to qualify the HL Loop mapping using multiple occurrence.

Mapping Hierarchical Loops



A window will display to allow you to open the current HL Loop mapping or create a new HL Loop mapping.

Mapping Hierarchical Loops

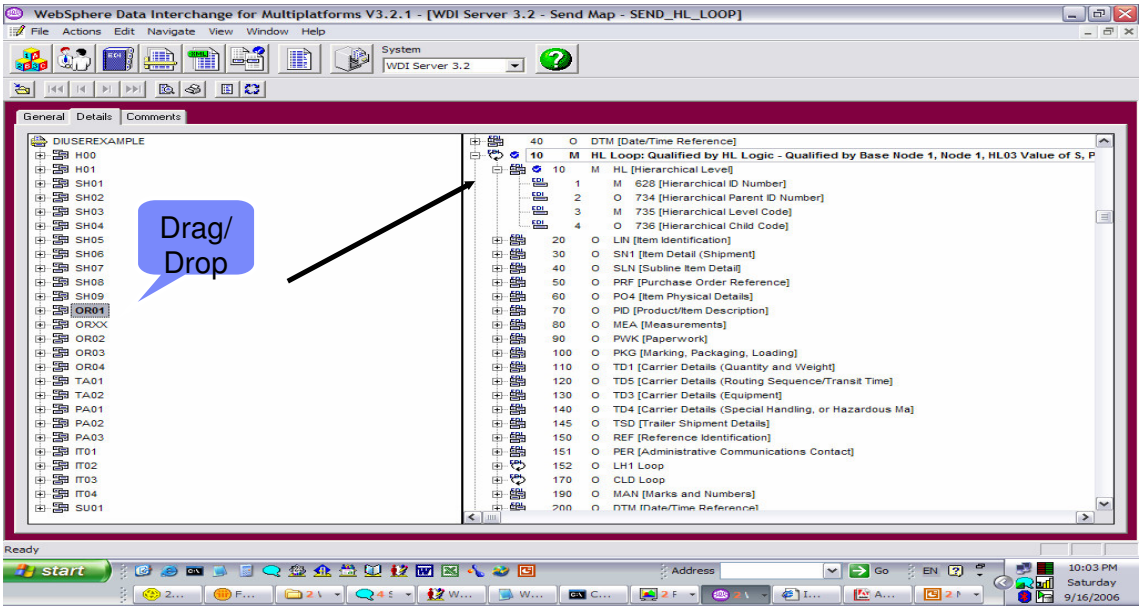


The Data Format Path field will be populated with the source element used for the drag and drop. A single occurrence mapping would not contain a value in the Data Format Path.

Section

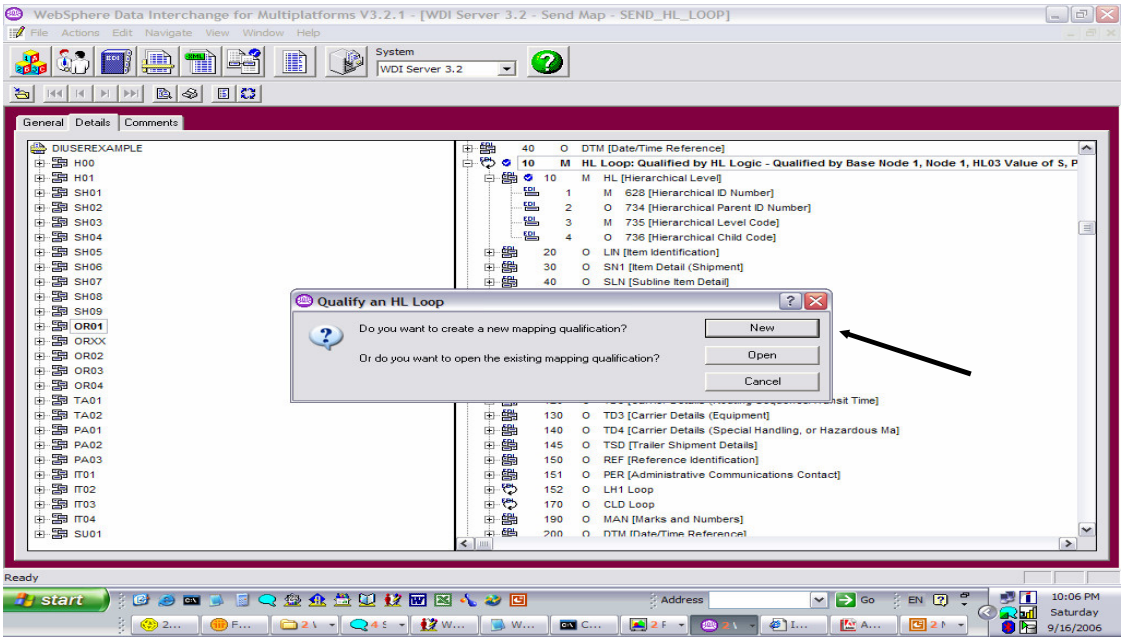
Adding parent/child relationship

Mapping Hierarchical Loops



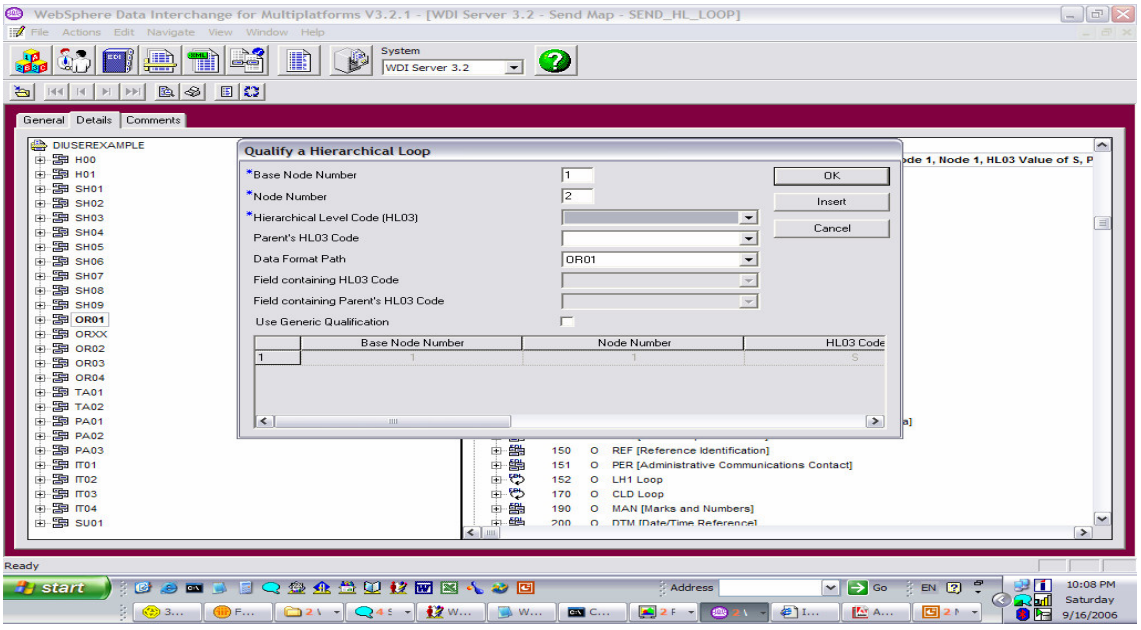
To create the underlying HL nesting levels or children for this HL level, drag and drop the next source element that will create the HL Loop level or double click on the HL Loop.

Mapping Hierarchical Loops



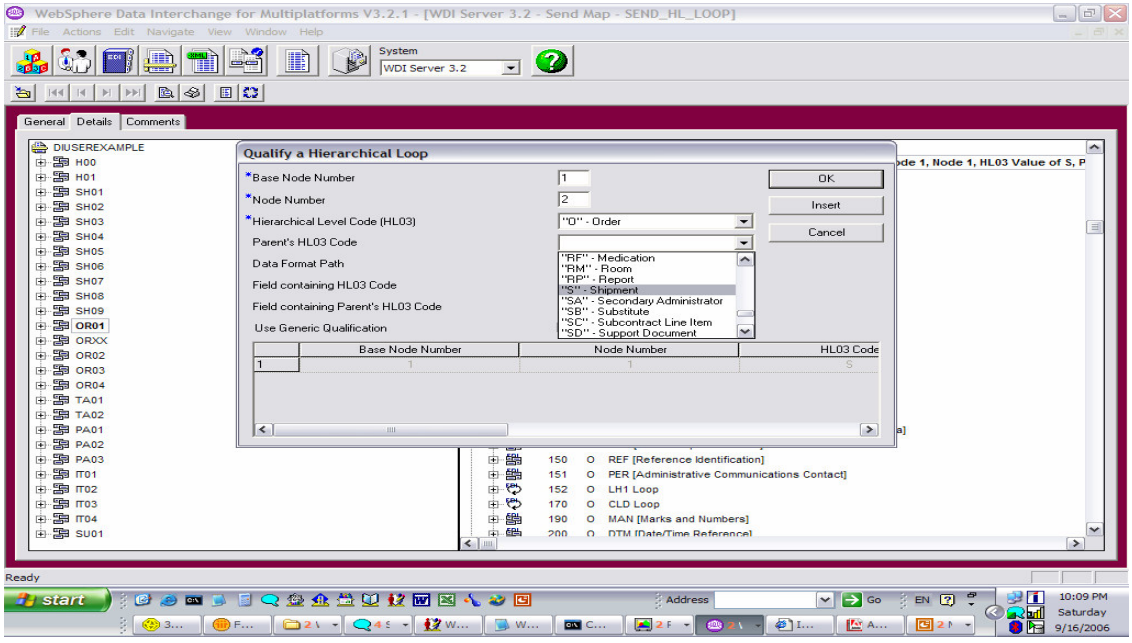
Select New.

Mapping Hierarchical Loops



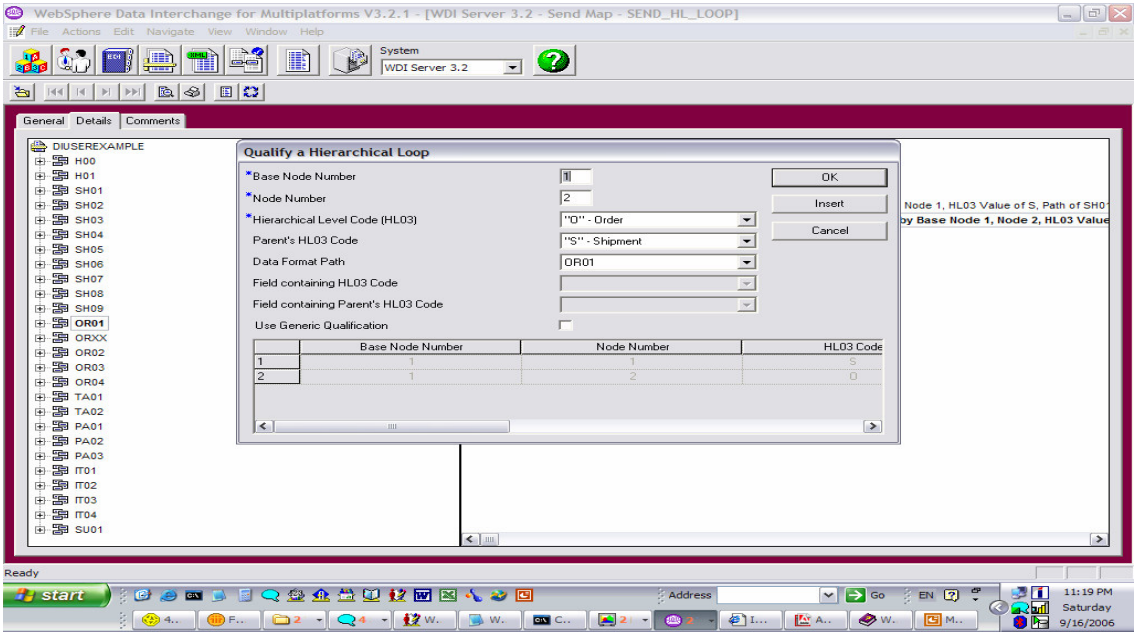
If this HL Loop mapping is part of the same base mapping change the Node Number to the next level. For example this is the Order level and the base mapping is for Shipment. The Order level is a child of the Shipment level. The Shipment level is defined with base node 1 and node number 2. The order level will have base node 1 and node number 2. Select the hierarchical level code from the list.

Mapping Hierarchical Loops



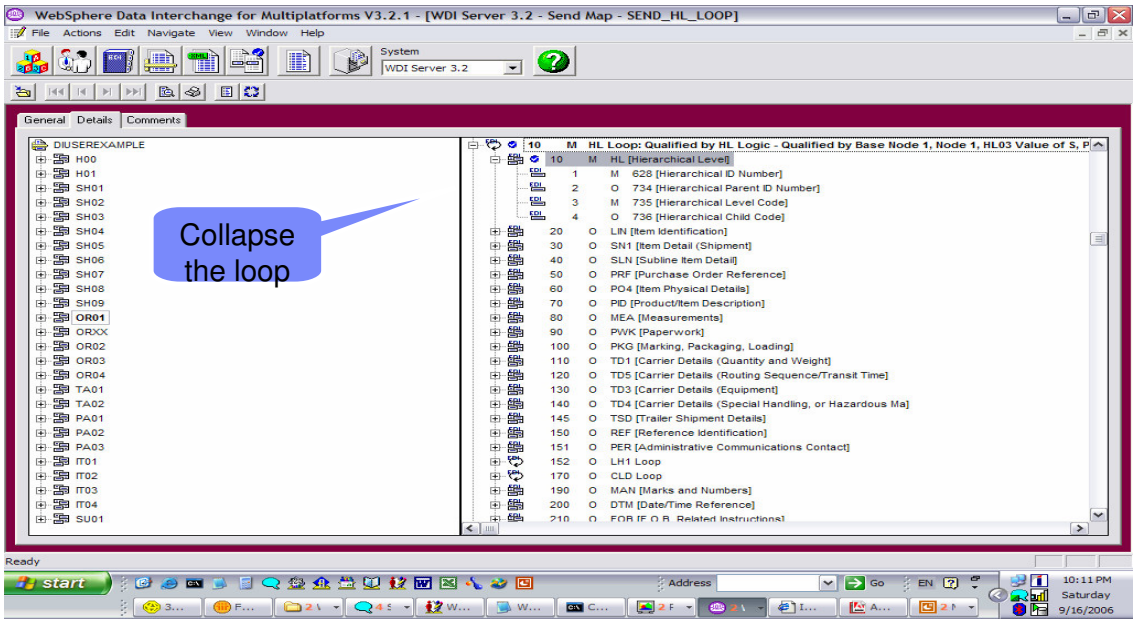
Select the hierarchical level code of the parent from the drop down list.

Mapping Hierarchical Loops



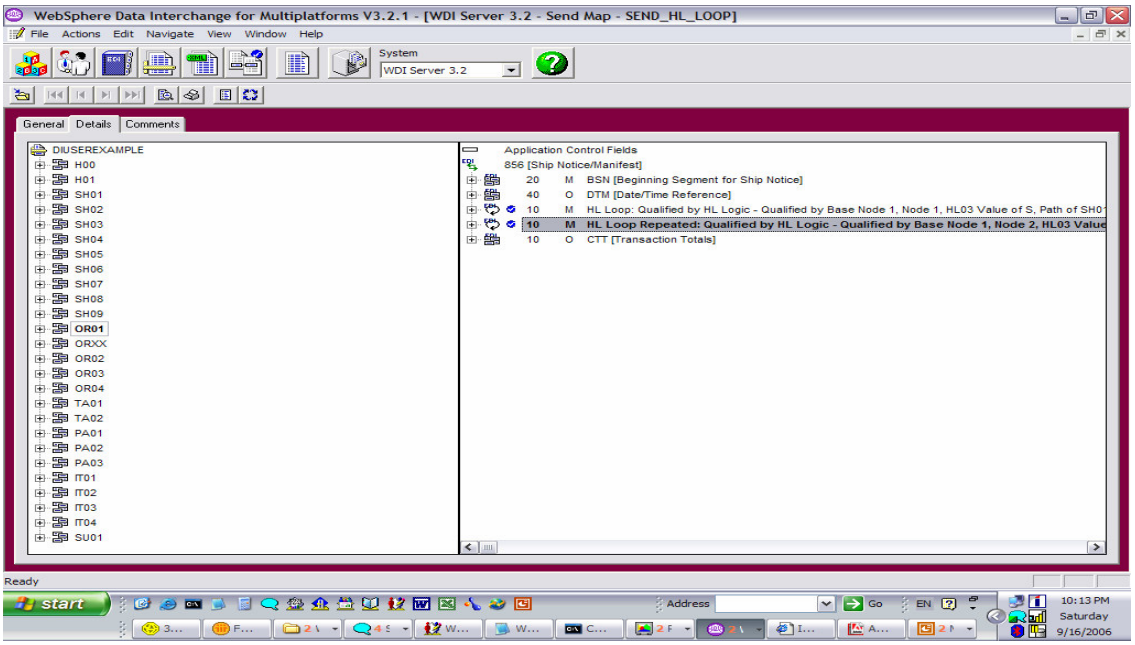
Identify the parent hierarchical level code. Order is a child of Shipment. When you are finished press OK.

Mapping Hierarchical Loops



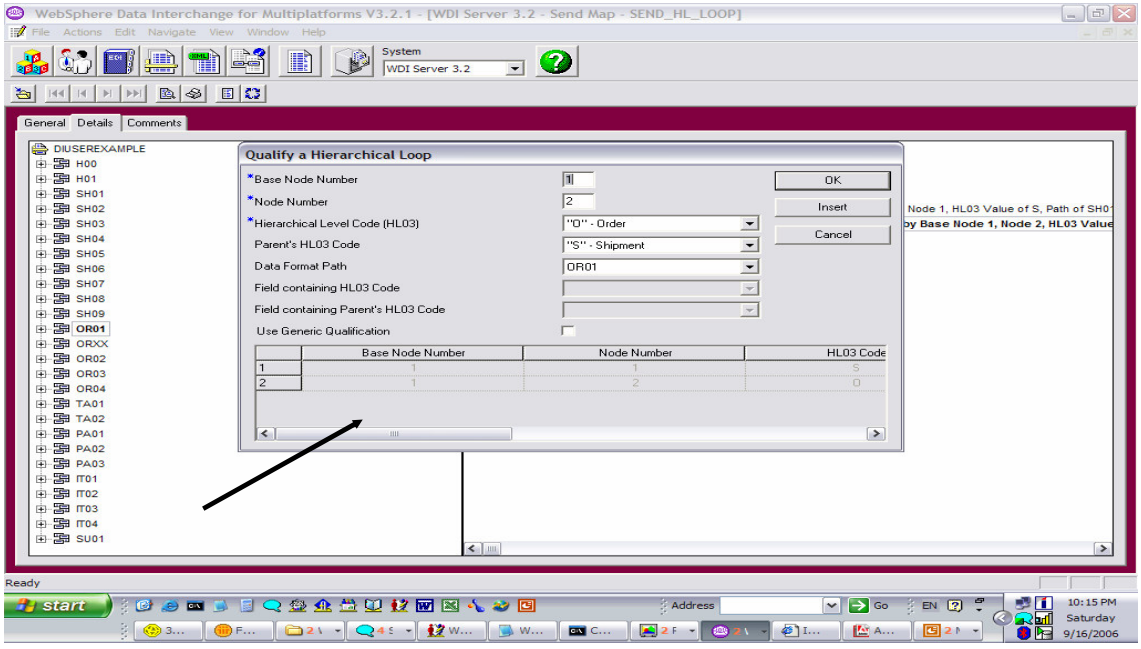
If you collapse the loop by clicking on the minus sign, you can see the new HL Loop qualification.

Mapping Hierarchical Loops



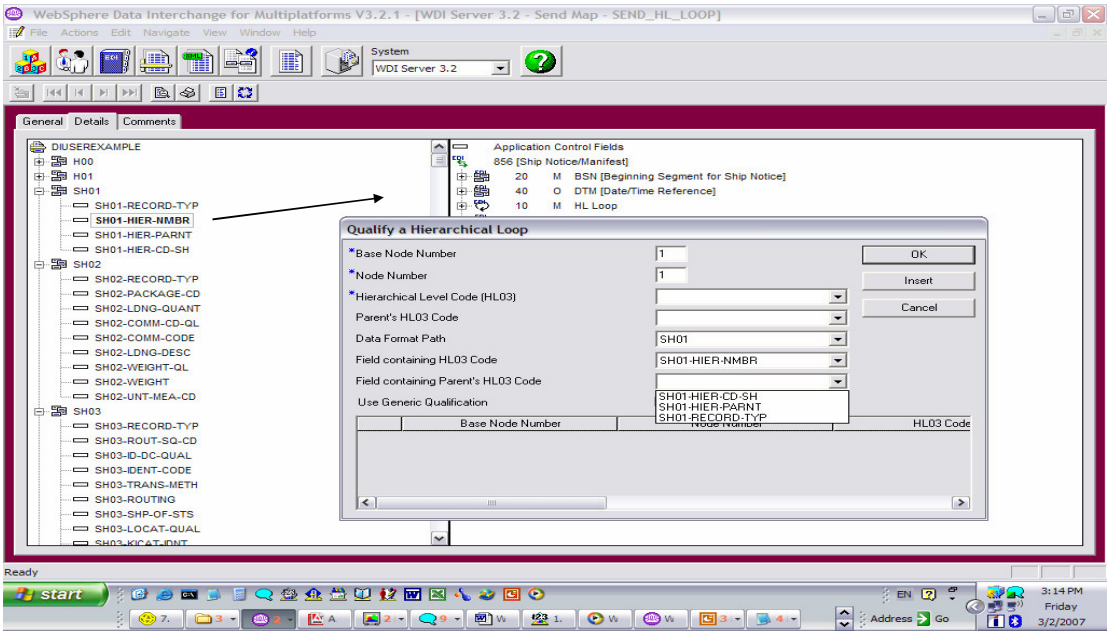
There are now 2 HL Loop qualifications.

Mapping Hierarchical Loops



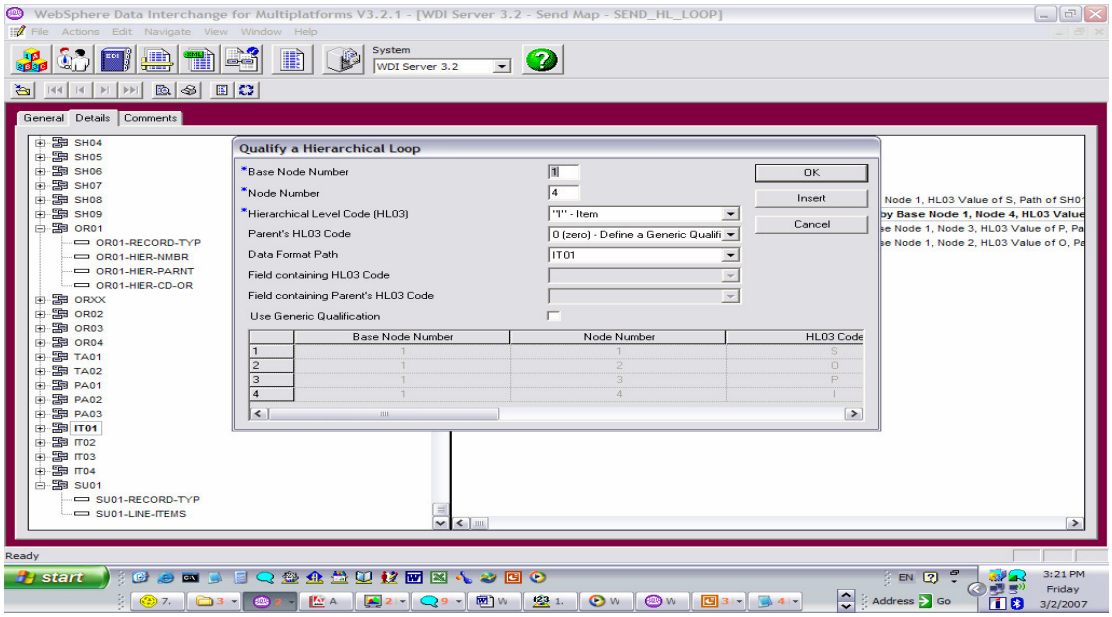
As you create the HL Loop mappings, the hierarchy is displayed at the bottom of the Qualify a Hierarchical Loop window.

Mapping Hierarchical Loops



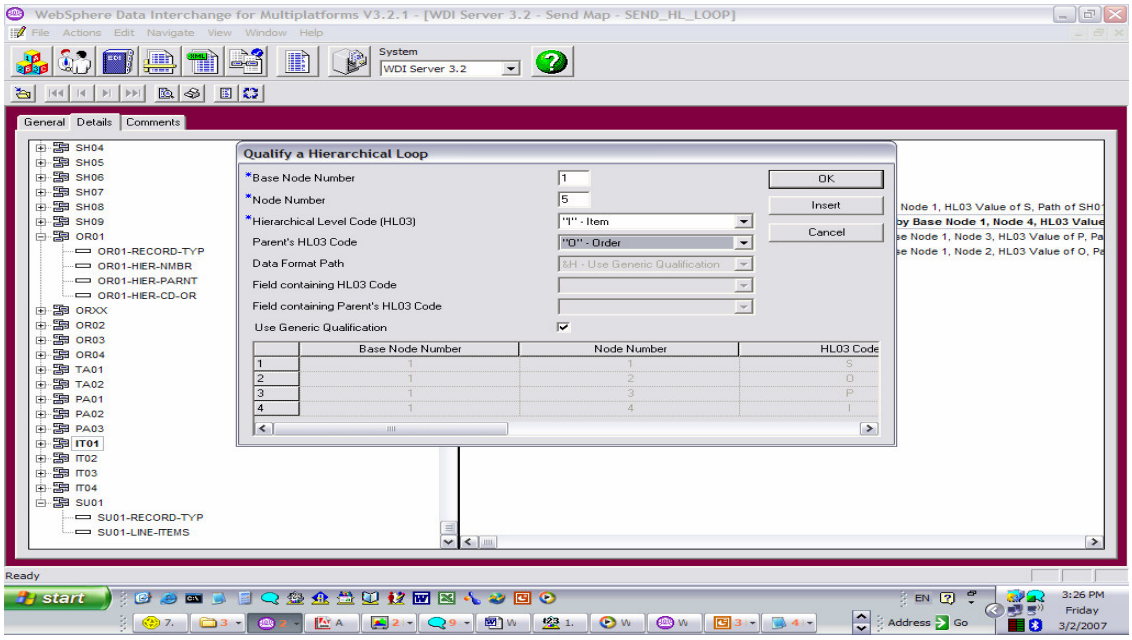
If you use a simple element in the source and drag and drop to the HL Loop, this will indicate the source data contains the HL Level code. You can then optionally select the field containing the parent's HL Level code.

Mapping Hierarchical Loops



A Blank in the Parent's HL Level code will define a generic mapping. Using a generic mapping you can map a level 1 time and identify this mapping is to be used with a parent child qualification by using the Use Generic Qualification flag.

Mapping Hierarchical Loops



With this qualification the Use Generic Qualification flag is checked the Data Format Path becomes &H indicating when the Item record follows an order in the source data to use the generic mapping. This qualification does not contain any other mapping commands because the commands are within the generic qualification. Use generic qualification to define parent-child relationships for which there are no specific mapping instructions. This means that the mapping is the same for this level and the parent does not affect the mapping

Section

Receive Mapping

Mapping Hierarchical Loops

- Two choices
- DE qualification on HL CODE - HL03 (SEGMENT HL, DE 735)
- Hierarchical Loop Support
  - Mapping dependent on parent level
    - Mapping may be different depending on the parent
  - Allow Multiple hierarchies in the same map
    - EXAMPLE: Orders may or may not have Shipment level

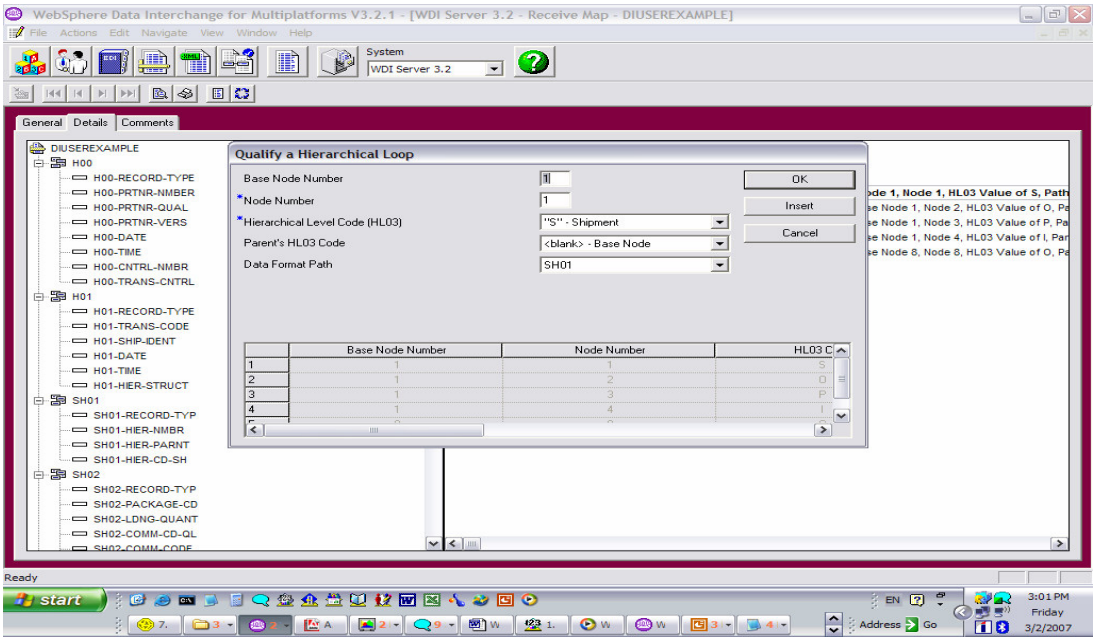
With a Receive map you have 2 choices for mapping and qualification. Use the Hierarchical Loop mapping support or use Qualify by value and normal mapping. You should use the Hierarchical Loop support if the mapping id dependent on which level is the parent or if you need to define different hierarchies within the same map. For example Orders without a Shipment level may be a different mapping than Orders with the Shipment level as the parent.

Mapping Hierarchical Loops

- The HL segment is the first segment of the HL loop
- HL*ID Number*Parent ID Number*Level Code*Child Code!
- Node ID
  - Identifies the mappings for a hierarchical loop.
  - Must be numeric.
  - HL segments define a top-down/left-right ordered structure.
- Hierarchical Level Code
  - The value in the HL03 field of the HL segment is compared to this value to determine which map to use.
- Parent Level Code
  - The hierarchical level code of the node to be the parent.
  - Zero indicates a generic mapping.
  - Blank indicates a base node for a hierarchical structure.

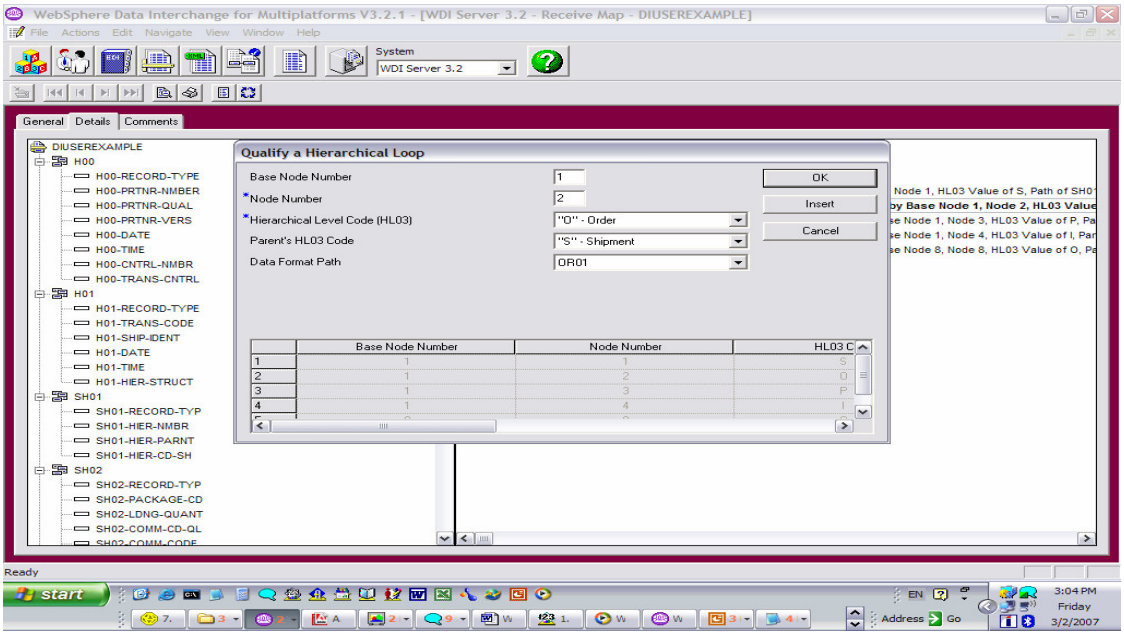
With a Receive map the value in the HL03 field of the HL segment is compared to the value in the map to determine which map qualification to use. This would be the same as using a Qualification using the value in the HL03 element without special HL Loop mapping. A parent level code of zero indicates a generic mapping. This means that the mapping is the same for this level and the parent does not affect the mapping. A blank in in the parent level code indicates a base node.

Mapping Hierarchical Loops



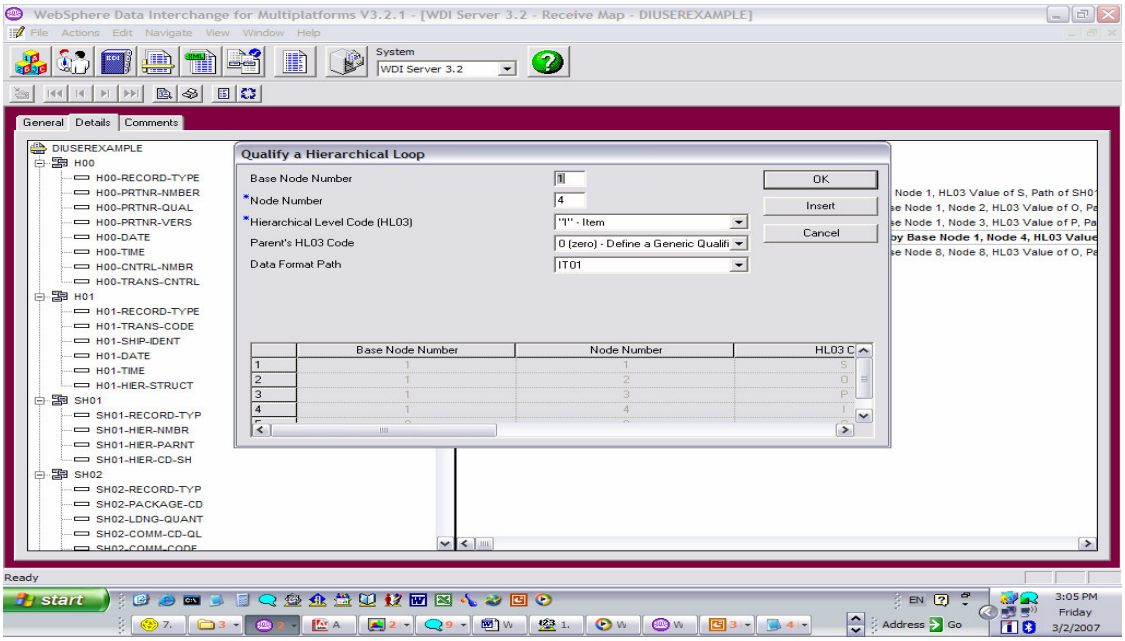
This qualification is defining the base node as Shipment.

Mapping Hierarchical Loops



This qualification is defining the Order level with a parent of Shipment.

Mapping Hierarchical Loops



This is a generic qualification because the parent level contains a value of zero. This indicates the Item level can follow or be a child of any other HL Level and the mapping should be the same regardless of the parent. Use generic qualification to define parent-child relationships for which there are no specific mapping instructions.

Summary

- Send and Receive mapping:
  - Allows you to specify unique mapping instructions for each identifiable group of structures in a hierarchical loop.
  - Can handle 16 levels of nesting within the HL loop structure.

Send and Receive mapping allows you to specify unique mapping instructions for each identifiable group of structures in a hierarchical loop. The translation process can handle 16 levels of nesting within the HL loop structure.

Reference

- More information can be found in the WDI V3.3 Mapping Guide, Appendix B. Hierarchical loops.

More information can be found in the WebSphere Data Interchange Version 3.3 Mapping Guide, Appendix B. Hierarchical loops.

Template Revision: 04/25/2006 11:09 AM

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.

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.