



IBM Software Group

## ***IBM WebSphere® Data Interchange V3.3***

### ***Using the XMLSPLIT feature***



@business on demand.

© 2007 IBM Corporation

This presentation will describe the XMLSPLIT feature.

## Agenda

- What is XML Split
- XML Split Setup



The presentation will describe the XML Split feature and setup.

## Section

# *What is XML Split*

## What is XML Split

- The default XML document processing - enables mapping a single XML source document to a single target document.
- Many XML source documents resemble Electronic Data Interchange (EDI) Standard data.
- The XML Split function will split a single XML document based on a defined XML compound element and reconstruct the document before the document enters the data transformation message flow.



The default XML document processing using the data transformation process enables mapping of a single XML document to a single target document. Many XML source documents resemble Electronic Data Interchange (EDI) Standard data. A single XML document contains header type information, multiple messages, and trailer type information. It is desirable for example to map the XML source document to multiple EDI target documents. To achieve this, a double transformation process is needed to transform the XML document to an intermediate document, for example data format, with a second transformation process to map the intermediate document to the EDI document.

The XML Split function will split a single XML document based on a defined XML compound element and reconstruct the document before the document enters the data transformation message flow.

## What is XML Split

- The following features enable the XML document split:
  - **Client** a overview tab allows selection for DTD definition fields. Sender/Receiver Qualifier/ID paths. Fields to identify how to split an XML document.
  - **PERFORM keywords** XMLSPLIT(Y/N) default is Y. WDI processing will ONLY split documents with DTD definitions that define the split tag definitions.
  - **Mapping properties.** InputMsgCnt (number of input messages processed with this map execution), LastMsg(Y/N) (Identifies that this is the last source message in the input file, and MsgSplitCnt (number of split documents from a single input XML doc).



In the WebSphere Data Interchange (WDI) Client XML functional area you can define trading partner information and identify how to split an XML document. With the XML editor, the Overview tab allows selection for DTD definition fields for example Sender/Receiver Qualifier/ID paths and the compound elements to identify the header, detail, and trailer to perform the split. There are PERFORM keywords that control if the XML document should be split.

WDI processing will ONLY split documents with DTD definitions that define the split tag definitions. There are 3 mapping properties that can be used during processing. InputMsgCnt is the number of messages processed. LastMsg will identify the last source input message. And MsgSplit identifies the number of split documents.

## XML Split Setup

- Three elements that may be defined to split the XML document.
  - ▶ Element identifying the header area in the XML document
  - ▶ Element identifying the individual messages (split area)
  - ▶ Element identifying the trailer area in the XML document.
- The element identifying the individual messages is **required** to split the source XML document. If the header area is not defined, the beginning of the XML document up to the element identifying the message will be used to construct a header area for the split document.
- If the trailer area is not defined, the end root element will be used as the trailer area for the split document.



There are three elements that may be defined to split the XML document. Element identifying the header area in the XML document, element identifying the individual messages (split area), and element identifying the trailer area in the XML document. The Trailer area identification may be a terminating element (end tag).

These definitions are used to split and reconstruct the XML documents before they are placed in the data transformation message flow.

The element identifying the individual messages is required to split the source XML document. If the element identification is not defined, the source XML document will not be split.

If the header area is not defined, the beginning of the XML document up to the element identifying the message will be used to construct a header area for the split document.

If the trailer area is not defined, the end root element will be used as the trailer area for the split document. If the trailer area is defined and is actually a terminating element in the XML source input, to define this using WDI Client is to right click the compound element (that begins the trailer element), define the element as the trailer element, using the check the box on the general tab "Element Terminator Indicates Start of Trailer Section".

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                     <=== Header Element Company 1
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
  </company>                                     (Note: end of header area)
  - <employee-list>
    + <employee>                                 <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
  </employee-list>                               <==== Trailer Element (Element Terminator)
+ <company>                                     <==== Company 2
+ <employee-list>
+ <company>                                     <==== Company 3
+ <employee-list>
</root-element>

```



The sample XML document below contains information about three companies. The company compound element is repeated 3 times and the first company is expanded. The expanded elements (those preceded by a dash) show the first company element contains information about the company (company-details) and information about four employees of the company within employee-list. If each employee element needs to be translated into its own document, then the "employee" element would be listed as the Message Element on the General tab page of the DTD/Schema Editor. The Header Element would be the "company" element. The Trailer Element would be the "employee-list" element with the Element Terminator Indicates Start of Trailer Section check box set.

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                     <=== Header Element Company 1
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
  </company>                                     (Note: end of header area)
  - <employee-list>
    + <employee>                                 <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
  </employee-list>                               <==== Trailer Element (Element Terminator)
+ <company>                                     <==== Company 2
+ <employee-list>
+ <company>                                     <==== Company 3
+ <employee-list>
</root-element>

```



The WDI Logical Message Adapter will parse 1 XML document from the input and pass this to the WDI message flow. The Logical Message Adapter will also perform the splitting of the XML document.



## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                     <=== Header Element Company 1
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
    </company>                                     (Note: end of header area)
  - <employee-list>
    + <employee>                                   <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
    </employee-list>                               <==== Trailer Element (Element Terminator)
  + <company>                                     <==== Company 2
  + <employee-list>
  + <company>                                     <==== Company 3
  + <employee-list>
</root-element>

```



The ?xml down to the header area is included with each split.

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                     <=== Header Element Company 1
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
    </company>                                     (Note: end of header area)
  - <employee-list>
    + <employee>                                     <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
    </employee-list>                               <==== Trailer Element (Element Terminator)
  + <company>                                     <==== Company 2
  + <employee-list>
  + <company>                                     <==== Company 3
  + <employee-list>
</root-element>

```



The data from the company start tag down to the company end tag is the first header area.

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                     <=== Header Element  Company 1
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
    </company>                                     (Note: end of header area)
  - <employee-list>
    + <employee>                                   <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
    </employee-list>                               <==== Trailer Element (Element Terminator)
  + <company>                                     <==== Company 2
  + <employee-list>
  + <company>                                     <==== Company 3
  + <employee-list>
</root-element>

```



The detail area for the first split will contain the first employee within the employee-list.

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                     <=== Header Element Company 1
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
    </company>                                     (Note: end of header area)
  - <employee-list>
    + <employee>                                   <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
    </employee-list>                               <==== Trailer Element (Element Terminator)
  + <company>                                     <==== Company 2
  + <employee-list>
  + <company>                                     <==== Company 3
  + <employee-list>
</root-element>

```



The trailer area is a terminating element for employee-list. The trailer area will be attached to the split document along with the end root element.

## XML Split Setup

```
<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>
    - <company-details>
      <company-name>Company 1 </company-name>
      + <company-address>
    </company-details>
  </company>
- <employee-list>
  + <employee>
</employee-list>
</root-element>
```



This is the resulting document from the first split. Since there are 4 employee elements and this is defined as the message element, there will be 4 splits or 4 XML documents constructed and passed to the WDI message flow. Each document will use the same company header area information for each employee.

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  + <company>                                <==== Company 1
  + <employee-list>
- <company>      <==== Header Element Company 2
  - <company-details>
      <company-name>Company 2 </company-name>
      + <company-address>
      </company-details>
  </company>                                (Note: end of header area)
- <employee-list>
      + <employee>                            <==== Message Element (Split here)
      + <employee>
  </employee-list>                          <==== Trailer Element (Element Terminator)
  + <company>      <==== Company 3
  + <employee-list>
</root-element>

```



With the remaining company elements, the header area containing the company information will be constructed using the information from the each company element, which would be the second and third company elements and the splitting will continue based on the number of employee elements following each company.

## XML Split Setup

```
<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>
    - <company-details>
      <company-name>Company 2 </company-name>
      + <company-address>
    </company-details>
  </company>
  - <employee-list>
    + <employee>
  </employee-list>
</root-element>
```



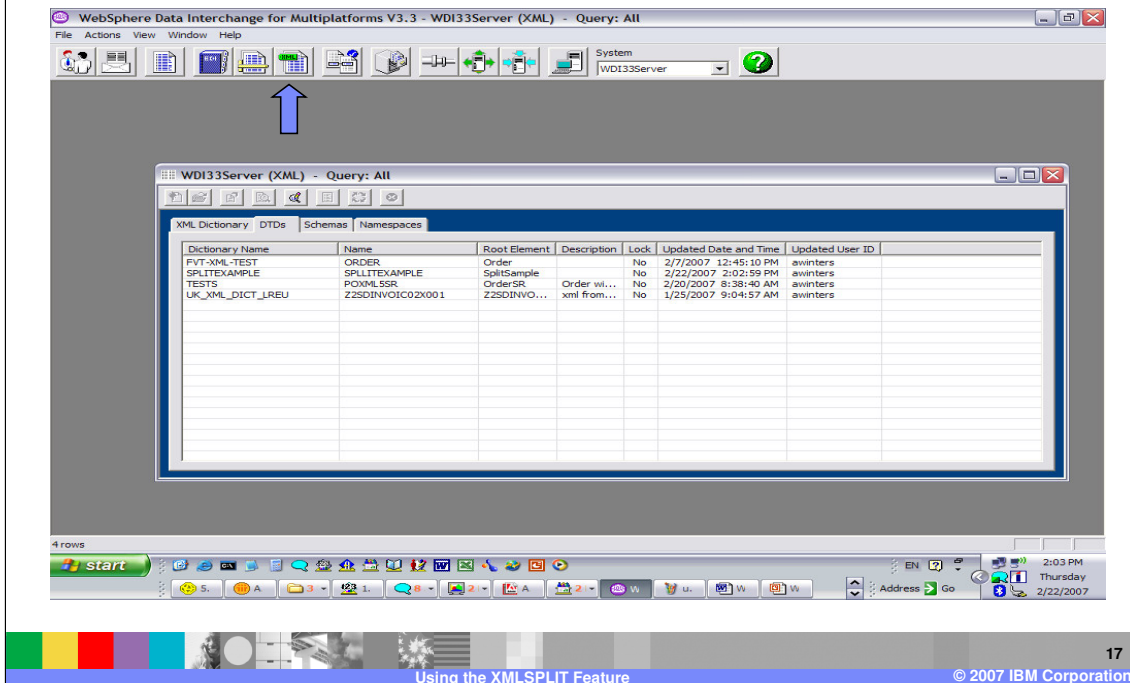
This is the resulting document from the split using the Company header information for company 2. Since there are 2 employee elements and this is defined as the message element, there will be 2 splits or 2 XML documents constructed and passed to the WDI message flow. Each document will use the same company header area information for each employee.

## Section

# *WDI Client Features*

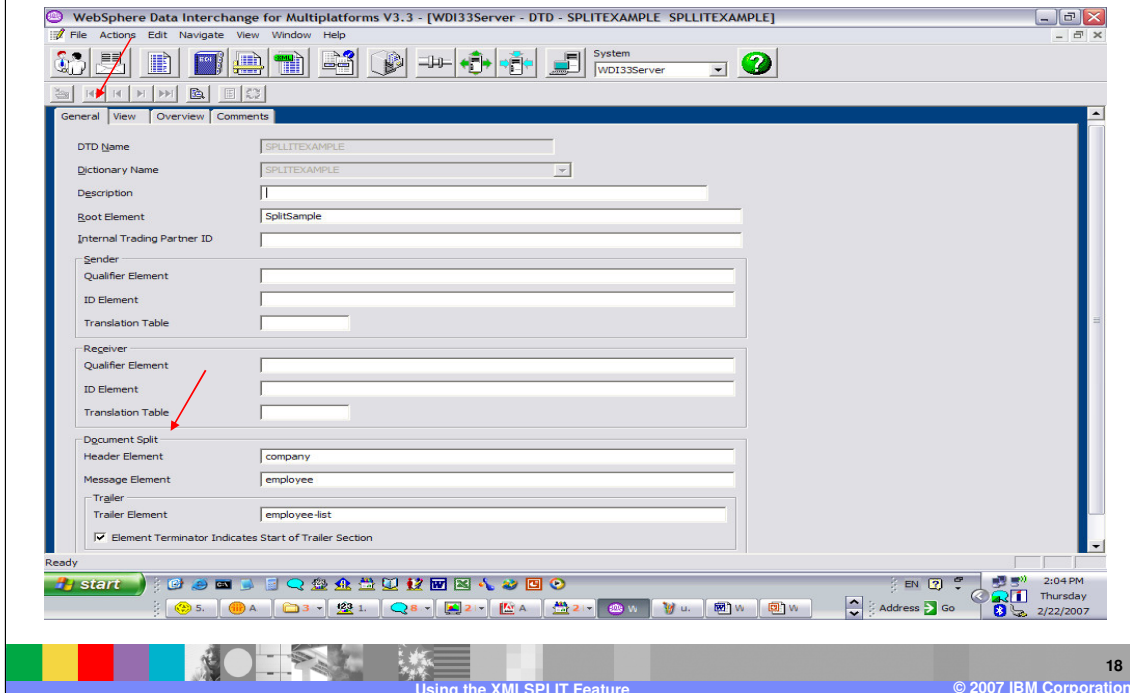


# XML Split Setup



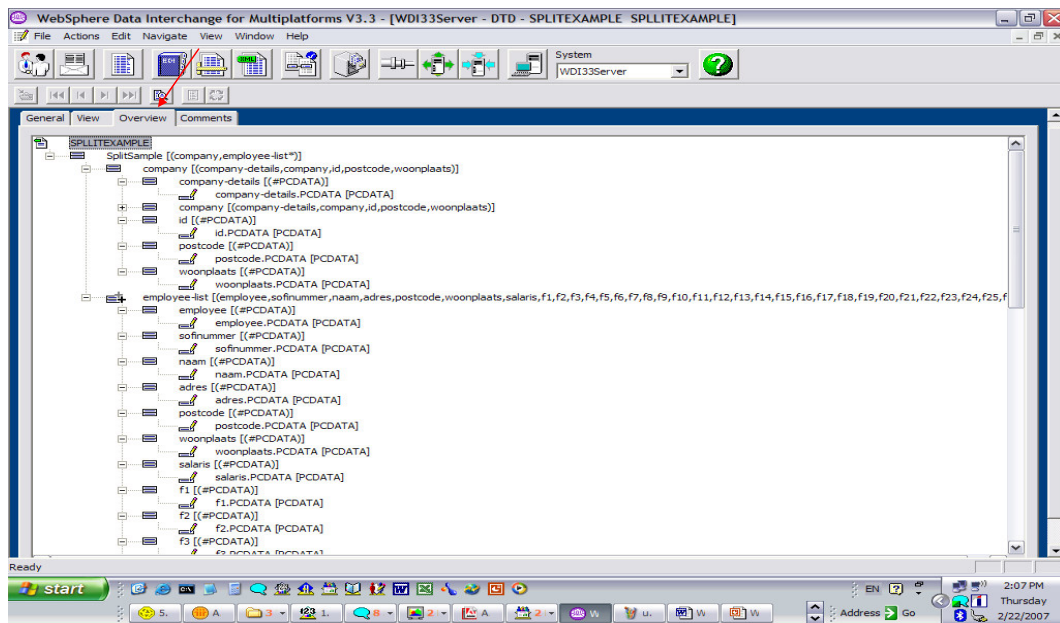
The XML Split setup is located in the XML functional area of WDI Client.

# XML Split Setup



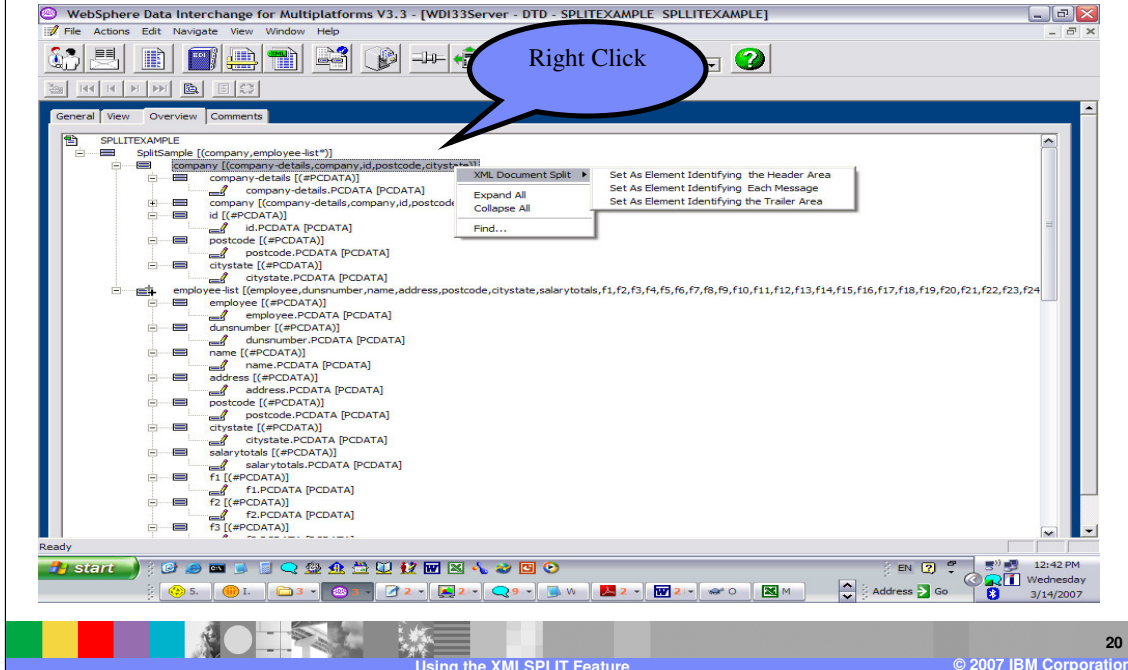
The XML Split setup is located under the Document Split section on the General Tab.

# XML Split Setup



To avoid typing mistakes, you can move to the Overview Tab and select the Document Split values.

# XML Split Setup



Right Click on the Compound element and select the XML Document Split. In our example, company is the Header area.

# XML Split Setup

The screenshot displays the WebSphere Data Interchange for Multiplatforms V3.3 interface. The main window shows a tree view of an XML document structure. The 'employee' element is selected, and a right-click context menu is open, highlighting the 'XML Document Split' option. A blue speech bubble with the text 'Right Click' points to the 'employee' element. The context menu options include: 'XML Document Split', 'Expand All', 'Collapse All', 'Find...', 'Set As Element Identifying the Header Area', 'Set As Element Identifying Each Message', and 'Set As Element Identifying the Trailer Area'. The interface also shows a taskbar at the bottom with the Windows Start button, system tray, and date/time (Wednesday 3/14/2007, 12:43 PM). The footer of the slide contains the text 'Using the XMLSPLIT Feature' and '© 2007 IBM Corporation'.


Employee is the element where we want to split the XML Document.

## XML Split Setup

```

<? xml version="1.0" encoding="UTF-8" ?>
- <root-element xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="ThisDoc.xsd">
  - <company>                                <=== Header Element
    - <company-details>
      + <company-name>
      + <company-address>
    </company-details>
  </company>                                (Note: end of header area)
  - <employee-list>
    + <employee>                             <==== Message Element (Split here)
    + <employee>
    + <employee>
    + <employee>
  </employee-list>                          <==== Trailer Element (Element Terminator)
  + <company>
  + <employee-list>
  + <company>
  + <employee-list>
</root-element>

```

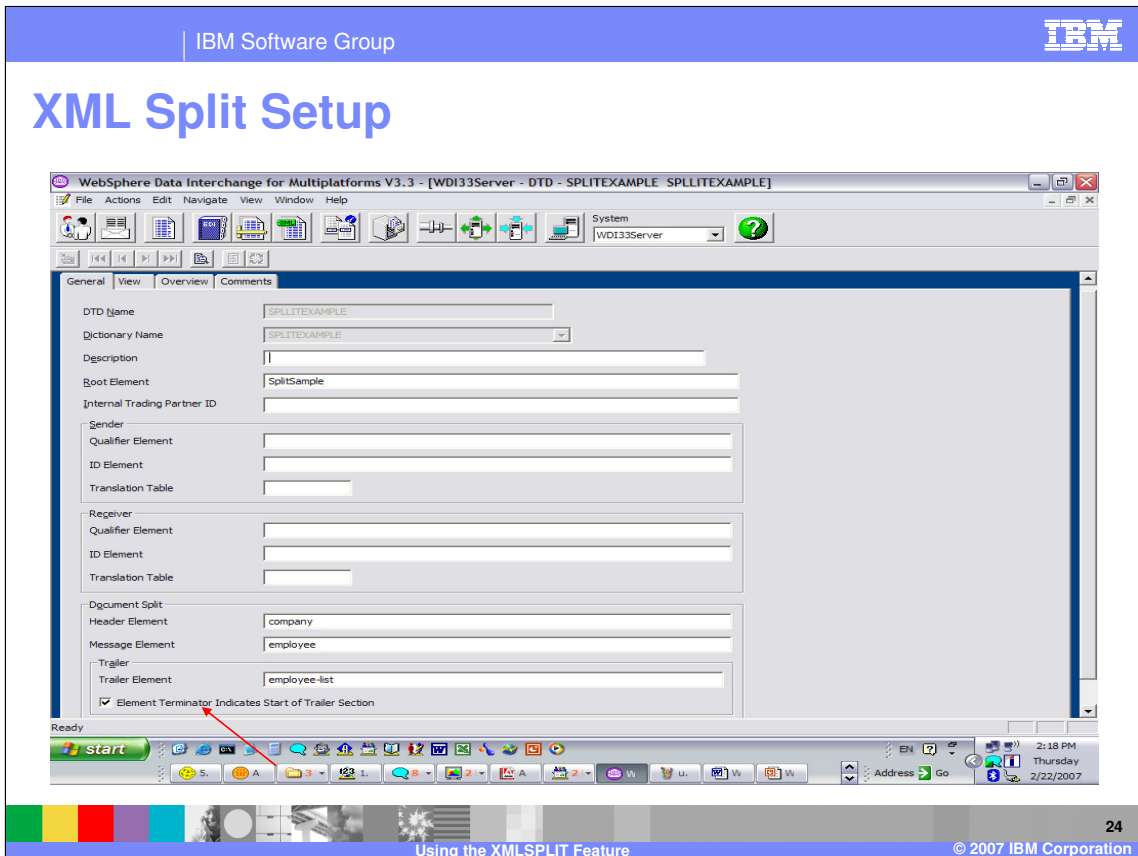



The element identifying the Trailer area is a terminating element.

# XML Split Setup

The screenshot displays the WebSphere Data Interchange for Multiplatforms V3.3 interface. The main window shows a tree view of an XML document structure. The root element is 'SplitSample', which contains two main sections: 'company' and 'employee-list'. The 'employee-list' section is selected, and a context menu is open over it. The menu options are: 'XML Document Split', 'Expand All', 'Collapse All', and 'Find...'. The 'XML Document Split' option is expanded, showing three sub-options: 'Set As Element Identifying the Header Area', 'Set As Element Identifying Each Message', and 'Set As Element Identifying the Trailer Area'. The 'Set As Element Identifying the Trailer Area' option is highlighted. The interface also shows a taskbar at the bottom with the Windows Start button, system tray, and a clock showing 12:45 PM on Wednesday, 3/14/2007. The page number '23' and the text 'Using the XMLSPLIT Feature © 2007 IBM Corporation' are visible at the bottom of the slide.

Identify the Trailer area. In our example the Trailer area is the terminating element for the employee-list element. The terminating elements are not displayed on the overview tab and are not selectable. To identify the terminating element we select the beginning element as the trailer area.

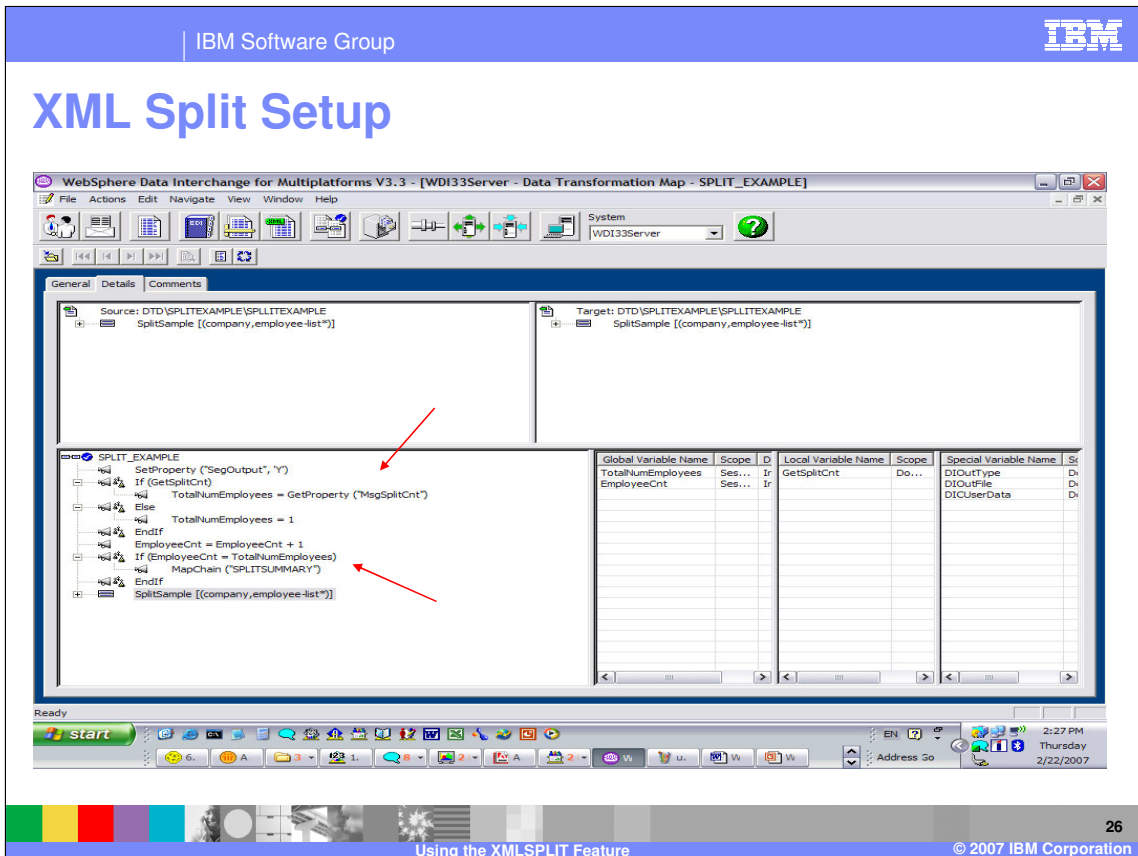


If we move back to the General Tab, the values selected on the Overview Tab have been set. If the element identifying the Trailer Area is a terminating element, the flag for this must be set.



## Section

# *WDI Client Mapping Properties*



In the Mapping Functional area, mapping properties can be used as controls for special processing. For example, if a summary map should be executed when the splitting has completed.

There are 3 mapping properties that can be used during processing. InputMsgCnt is the number of messages processed. LastMsg will identify the last source input message. And MsgSplit identifies the number of split documents.

## XML Split Setup

- Splitting is not possible with some encodings.
  - For z/OS the XML data must be some form of EBCDIC
  - For Windows/AIX it must be some form of ASCII
  
- XML input with encoding like UTF-16 should specify XMLSPLIT(N) on PERFORM TRANSFORM command.
  
- NO XML splitting desired should specify XMLSPLIT(N) on PERFORM TRANSFORM command.
  
- The XML work file XMLWORK must be allocated.



Since the Logical Message Adapter (LMA) is doing the XML splitting based on element names, Splitting is only possible for encodings native to the operating system. The LMA currently does not do any code page conversions for the split feature. XML input with encoding like UTF-16 should specify XMLSPLIT(N) on the PERFORM TRANSFORM command. This is because the Root element is extracted from the XML input data and a DTD/Schema lookup is done from the LMA to identify XML split element names. The XML work file must be allocated. This work file holds the re-constructed messages with each split. If you are not using XML split, you may want to specify XMLSPLIT(N). This bypasses the logic to scan the data and check the database to see if the Document split fields are defined for the DTD or schema. Skipping this check may slightly improve performance.

## Summary

- WDI provides a feature to allow a single XML document to be split
- The Client XML editor allows selection for the XML elements to control the split
- Mapping properties are available to control additional processing
- There are encoding restrictions for the split execution
- XMLSPLIT keyword can be used to turn splitting off



Options exist to split a single XML document based on a defined compound XML element and reconstructed before the document enters the Data Transformation message flow. There are restrictions on using the split function. The XMLSPLIT keyword can be used on the PERFORM command to turn off any processing for the XML split.

## 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  
IBM (logo)  
e/Logo/business  
AIX

CICS  
Cloudscape  
DB2  
DB2 Universal Database

IMS  
Informix  
iSeries  
Lotus

WMO  
OS/390  
OS/400  
pSeries

Tivoli  
WebSphere  
xSeries  
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 NON-INFRINGEMENT. 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.