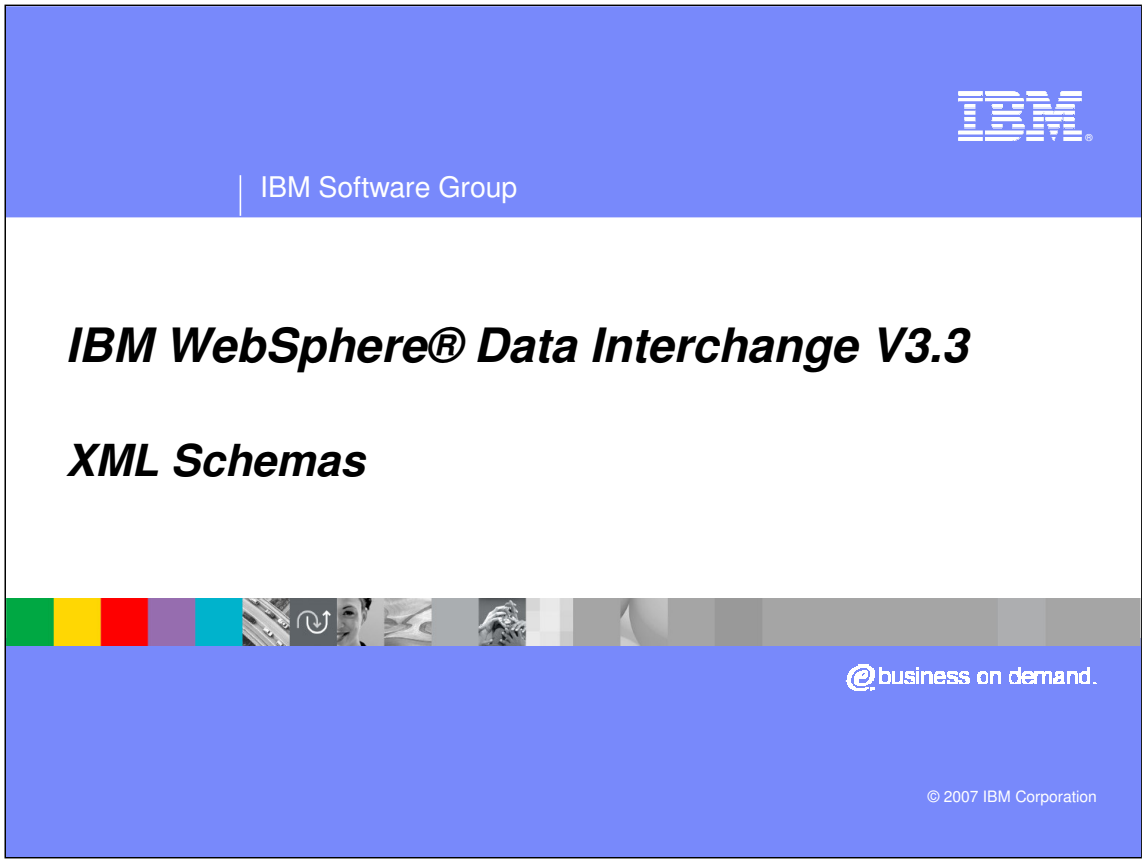


This presentation will review XML Schemas.

Agenda

- Discuss special considerations for using XML schemas with WDI
  - Show special XML mapping commands
  - Describe special XML PERFORM keywords

© 2007 IBM Corporation

The presentation will discuss some special considerations that you should be aware of when using XML schemas with WebSphere Data Interchange (WDI). There are some mapping commands and PERFORM TRANSFORM keywords that apply to schemas, but are not typically used with DTDs.

Schemas

- Like DTDs, XML schemas describe the structure of the XML document, but in much more detail
  - Allow constraints on elements and attributes (i.e., date, numeric, list of values, mask, etc.)
  - Min/max repeat counts
- Other functions and constructs that are not supported in DTDs
  - “all” content spec
  - Ability to define your own types, including base and derived types

© 2007 IBM Corporation

Like DTDs, XML schemas describe the structure of an XML document. However, they allow you to describe the data and semantics in much greater detail than you can in a DTD. XML schemas can define constraints like whether an element contains a date, integer, enumerated value, or even a specific pattern. Minimum and maximum repeat counts can also be defined. Some constructs such as the “all” content specification do not have an equivalent in DTDs.

Schemas are very flexible, too. You can define your own types, including base and derived types.

Because XML schemas can define the data structure more precisely, they are used to define many of the newer XML standards.

Mapping and translation using a schema

- Similar to using DTDs
  - Import the schema
  - Create the map
  - Run the TRANSFORM command
- A couple new mapping commands and TRANSFORM keywords to be aware of

© 2007 IBM Corporation

With WDI, mapping and translation for a document that is defined by an XML schema is similar to one defined by a DTD. The steps are still:

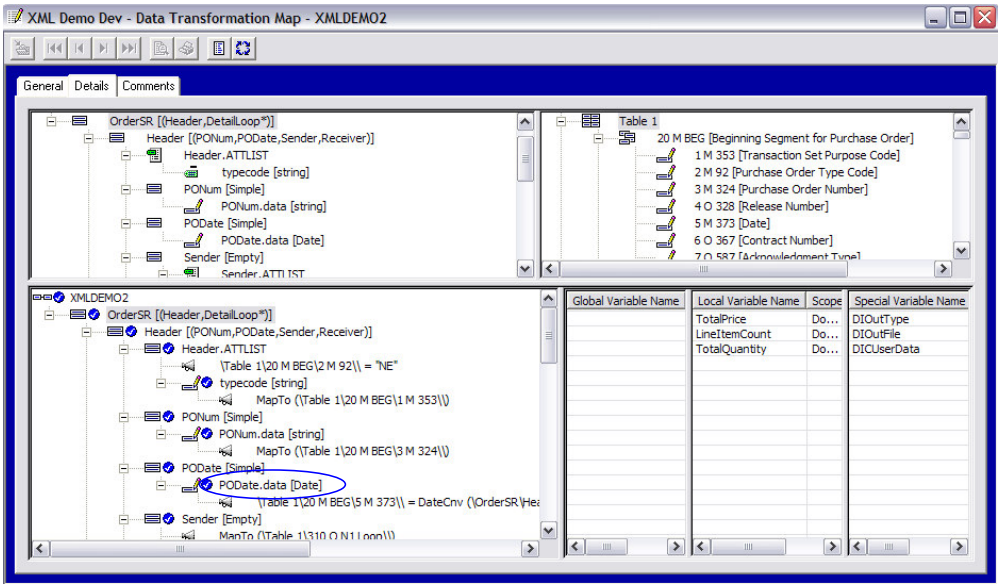
▪Import the schema

-Create the map

-Run the PERFORM TRANSFORM command

However, there are some new mapping commands and PERFORM TRANSFORM keywords that apply to schemas, but generally not to DTDs.

Schema map – source document



© 2007 IBM Corporation

Mapping from a source document defined by an XML schema is very similar to mapping from a document defined by a DTD.

The tree view in the map editor looks a lot like it did when using a DTD. One difference you may notice is that when the XML elements appear in the tree view, the specific types from the XML schema such as string, date, integer, etc. is displayed. With a DTD, these would just be identified as #PCDATA, since the DTD does not allow you to define the more specific types.

Special XML keywords for TRANSFORM from XML schema

- XMLNS(Y) – Namespace processing on
  - Suggest always using for schema source documents, even if no target namespace
  - Schemas themselves use namespaces
- Schemas used for validation also go in the directory specified by the XMLDTDs keyword

© 2007 IBM Corporation

The XMLNS PERFORM keyword tells WDI whether it should recognize namespaces when parsing the source document. When the source document is based on an XML schema, you should always specify XMLNS(Y). Even if the schema does not define a target namespace, schemas themselves use namespaces.

Like DTDs, if you are using the XML schema to validate the XML document, you need to put the XML schema file in the directory or partitioned data set indicated by the XMLDTDs keyword.

XMLSCHEMAVAL keyword

- XMLSCHEMAVAL(Y/N/A) – Controls schema validation
  - Y – Always try to do schema validation
  - N – Never do schema validation
  - A – Do schema validation if a schema location is specified. Otherwise, do not do schema validation
  - Note: Y or A will also do DTD validation if a DTD is specified in the data

© 2007 IBM Corporation

The XMLSCHEMAVAL PERFORM keyword tells WDI whether it should validate the document against the XML schema. This is separate from the XMLVALIDATE keyword, which controls DTD validation.

If you do XML schema validation, any DTDs that are referenced will also be processed. In other words, if the document or XML schema refers to a DTD, you cannot tell WDI to validate against the XML schema but ignore the DTD reference.

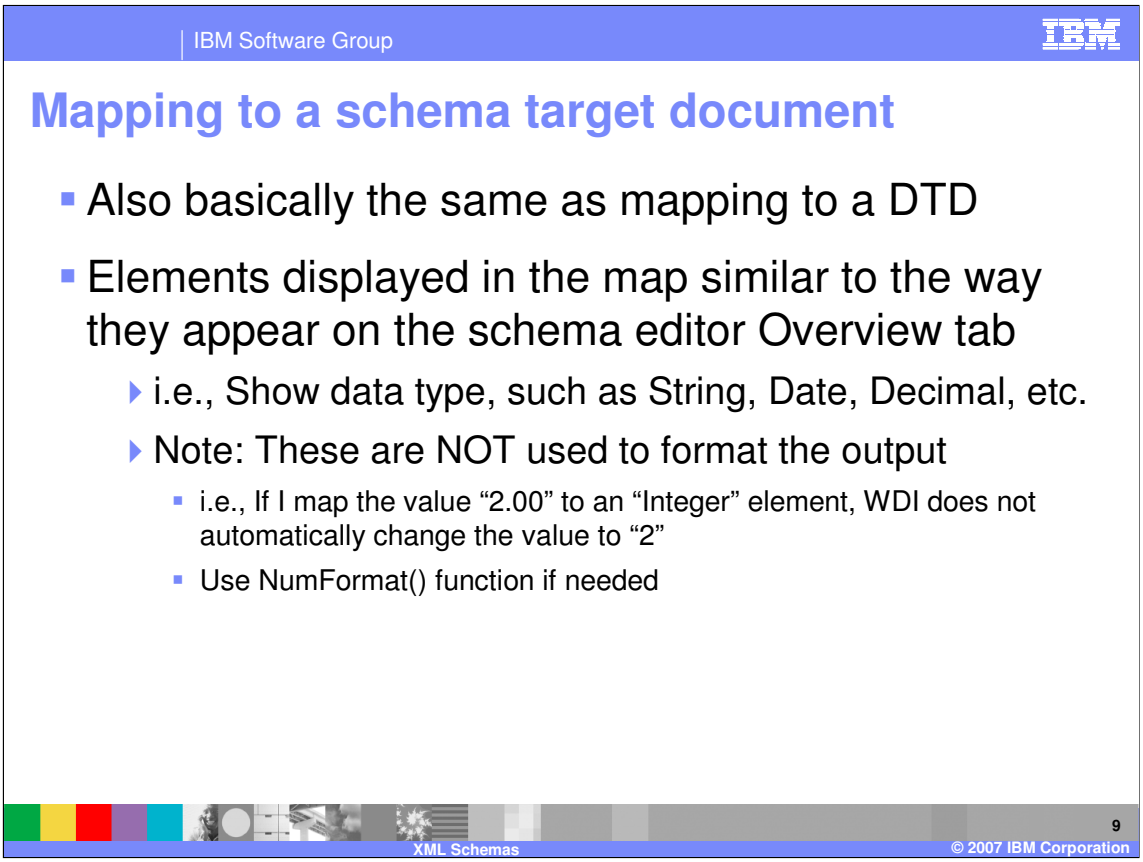
If you want, you can do DTD validation or processing by specifying the XMLVALIDATE keyword, but skip the XML schema validation.

Another PERFORM TRANSFORM example

```
PERFORM TRANSFORM WHERE
        INFILE(XMLFILE)
        SYNTAX(X)
        OUTFILE(EDIFILE)
        CLEARFILE(Y)
        XMLVALIDATE(2)
        XMLDTDS(\DEMO\DTDS)
        XMLSPLIT(N)
        DICTIONARY(XMLDEMO)
        DOCUMENT(EXAMPLE2)
        XMLNS(Y)
        XMLSCHEMAVAL(Y)
```


© 2007 IBM Corporation

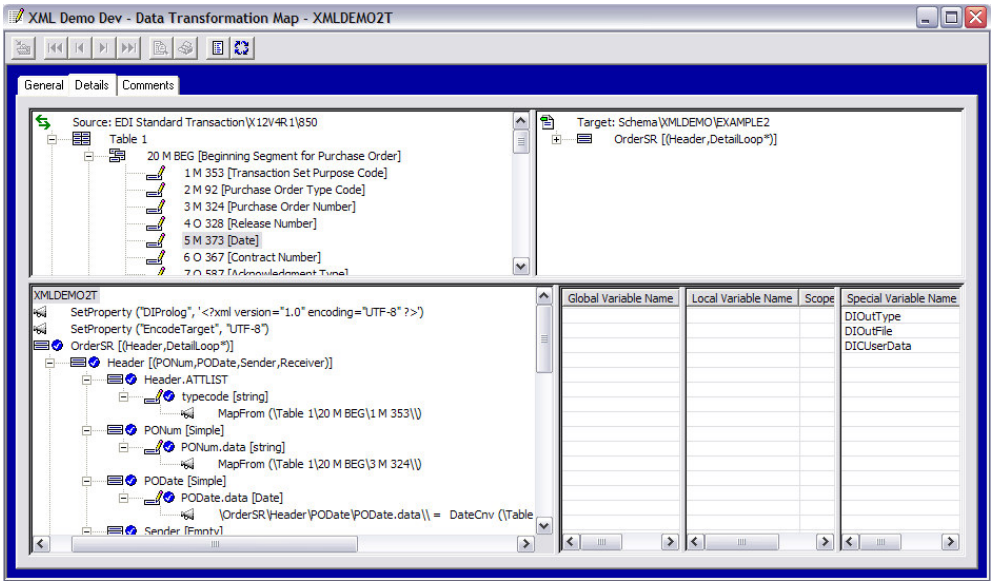
Here is a sample TRANSFORM COMMAND that uses namespace processing and will validate the data against the XML schema file that is located in the \DEMO\DTDS directory.



If you are mapping to a target document that is defined by an XML schema, this is similar to mapping to a target document defined by a DTD. Like we saw with the source document, the elements show the data type such as string, date, decimal, etc.

Note that WDI does not do any special formatting for the XML output based on the data types. For example, if you map a value of “2.00” to an element that is defined as an integer, WDI does not automatically change the value to “2”. However, WDI provides many mapping functions that allow you to format the output values as needed. In this example, you could use the NumFormat mapping command to convert the “2.00” to a “2”.

Schema map – target document



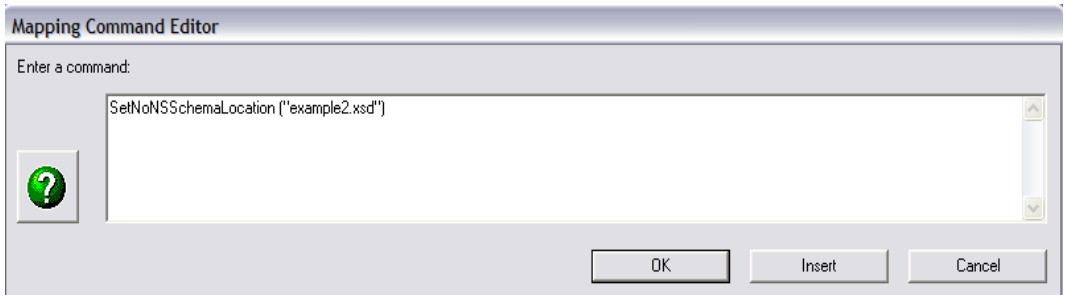
© 2007 IBM Corporation

Here is an example of a map that where the target document is based on an XML schema. Note that this is a target-based map, so the mapping commands appear relative to the elements in the target document.

SetNoNSNSchemaLocation command

- SetNoNSSchemaLocation
  - Creates the noNamespaceSchemaLocation attribute on root element

<OrderSR
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="example2.xsd">



© 2007 IBM Corporation

The SetNoNSSchemaLocation mapping command is used to create the noNamespaceSchemaLocation attribute on the root element. You would use this when the target document is based on a schema, you want to identify the schema name in the XML output, and the XML schema does NOT use a target namespace.

If you want to create the schemaLocation attribute for a schema that DOES use a target namespace, you use the SetSchemaLocation mapping command. This is covered in the XML Namespaces presentation.

Keywords for TRANSFORM to XML schema

- Again, no special PERFORM keywords for transforming to an XML schema-based target document
- Target output processing determined by map, rules, trading partner info, etc.

© 2007 IBM Corporation

Just like when you generate XML output data that is based on a DTD, there are no special PERFORM TRANSFORM keywords needed to generate output that is based on an XML schema. The target output processing is determined by the map, rules, trading partner information, etc.

Summary

- Mapping with XML schemas similar to DTDs
- Some special considerations
  - Special XML PERFORM keywords
  - Special XML mapping commands

© 2007 IBM Corporation

As you can see, using XML schemas with WDI is very similar to using DTDs. However, there are a few considerations to keep in mind, including some specialized PERFORM TRANSFORM keywords and mapping commands.

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.

© 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.