



IBM Software Group

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

## ***Code Lists***



@business on demand.

© 2007 IBM Corporation

This presentation will review Code Lists for Validation.

## Agenda

- Purpose of Code Lists
- Accessing Code Lists
- Mapping example
- Data Format example
- Restrictions
- Internals
- Summary



Topics to be covered in this presentation are:

What is the purpose of WebSphere Data Interchange (WDI) Code Lists?

How are Code Lists accessed?

What is an example of using a Code List in WDI Mapping?

What is an example of specifying a Code List in a Data Format?

Are there any Restrictions?

## Purpose of Code Lists

- Code Lists / Validation tables
  - ▶ Used during translation
  - ▶ Defined during mapping or Data Format (DF) Field definition
  - ▶ Verifies whether the value of an element is valid
  - ▶ Used automatically during EDI Syntax validation



Code Lists are sometimes referred to as Validation Tables. They are used to validate a value in an Electronic Data Interchange (EDI) standard element or Data Format Field. EDI Standards have code lists built into the standard. Part of the EDI Syntactical validation is to check to see if values in incoming and outgoing data conform to the values in the code list. The result of the validation is an indication that incoming data is “bad”. This gives the mapper an opportunity to alert users about the message or to take some corrective action. Additional validation can be performed with validation maps, and code lists can be used to verify both EDI and non-EDI data.

## Accessing Code Lists

- From the Standards Functional Area
  - ▶ Click on the Code Lists tab
  - ▶ Table can be named for reference during mapping

WebSphere Data Interchange for Multiplatforms V3.3 - Development (EDI Standards) - Query: All

Development (EDI Standards) - Query: All

EDI Standard Dictionary | Transactions | Segments | Data Elements | Code Lists | Envelope Standards | Envelope Control Strings

Code List Name	Description	Lock	Updated Date and Time	Updated User ID
0001 EDI	Syntax I...	No	10/12/2006 2:48:56...	admin
0007 EDI	Identific...	No	10/12/2006 2:48:56...	admin
0007D99A	Identific...	No	10/12/2006 2:48:56...	admin
0013D99A	Segmen...	No	10/12/2006 2:48:56...	admin
0025 EDI	Recipien...	No	10/12/2006 2:48:56...	admin
0029 EDI	Processi...	No	10/12/2006 2:48:57...	admin
0031 EDI	Acknowl...	No	10/12/2006 2:48:57...	admin
0035 EDI	Test ind...	No	10/12/2006 2:48:57...	admin
0051 EDI	Controll...	No	10/12/2006 2:48:57...	admin
0051D99A	Controll...	No	10/12/2006 2:48:57...	admin
0052 EDI	Messag...	No	10/12/2006 2:48:57...	admin
0052D99A	Messag...	No	10/12/2006 2:48:58...	admin
0054 EDI	Messag...	No	10/12/2006 2:48:58...	admin
0054D99A	Messag...	No	10/12/2006 2:48:58...	admin
0065 EDI	Messag...	No	10/12/2006 2:48:58...	admin
0065D99A	Messag...	No	10/12/2006 2:48:59...	admin
0073 EDI	First/As...	No	10/12/2006 2:48:59...	admin
0083D99A	Action...	No	10/12/2006 2:48:59...	admin
0085D99A	Error ty...	No	10/12/2006 2:48:59...	admin
0113 EDI	Messag...	No	10/12/2006 2:48:59...	admin
0133 EDI	Charact...	No	10/12/2006 2:49:00...	admin
143_UC5	Transac...	No	10/12/2006 2:49:00...	admin
143_X12	Transac...	No	10/12/2006 2:49:00...	admin
143U33	Transac...	No	10/12/2006 2:49:01...	admin
143X350	Code un...	No	10/12/2006 2:49:01...	admin
143X370	Transac...	No	10/12/2006 2:49:02...	admin
143X420	Transac...	No	10/12/2006 2:49:02...	admin

WDI Code Lists © 2007 IBM Corporation

Code Lists are located in the Standards and Mapping Functional Areas. Open the Standards Functional Area window by pressing the Standards button on the WebSphere Data Interchange Client Navigator bar. There you will find the Code Lists List window. The list window is used to list and perform maintenance functions on Code Lists.

## Accessing Code Lists

- Open a code list editor
  - ▶ Double click on a table name
- Each code list can have its own field size
- Values must be unique
- Validate mapping command uses a code list

The screenshot shows the 'Development - Code List - 0054\_ED1' window. It has two tabs: 'General' and 'Comments'. The 'General' tab is active and contains the following fields:

- Name: 0054\_ED1
- Description: Message type release number
- Data Type: Character (dropdown menu)
- Maximum Length: 3

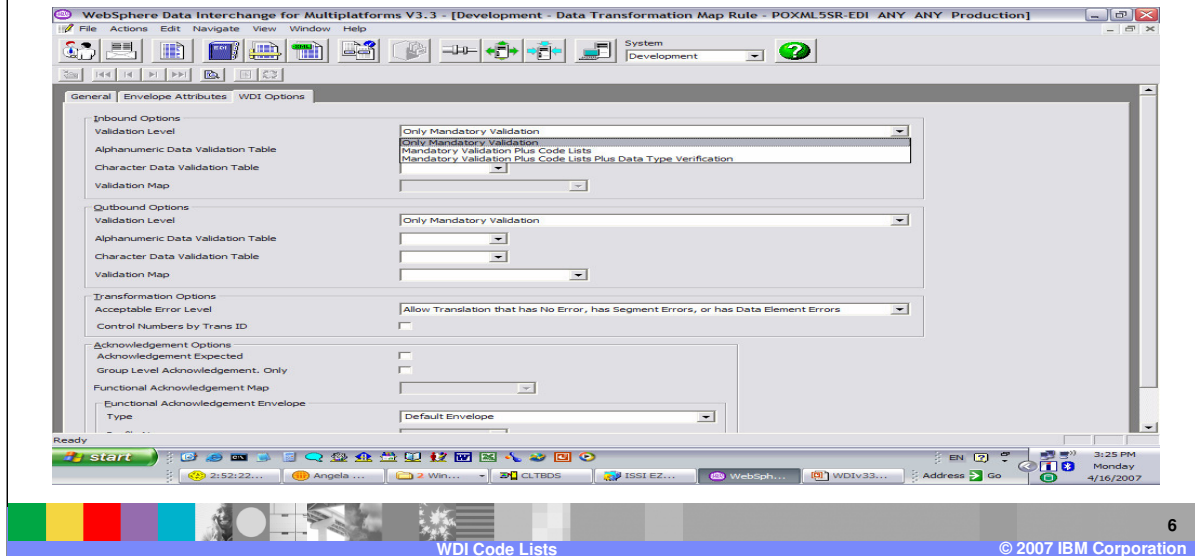
There are three buttons on the right: 'New...', 'Open...', and 'Delete'. Below the form is a table with two columns: 'Value' and 'Description'.

Value	Description
1	First rel...
2	Second ...
902	Trial rele...
911	Trial rele...
912	Standar...
921	Trial rele...
932	Standar...
93A	Release ...
94A	Release ...
94B	Release ...
95A	Release ...
95B	Release ...
96A	Release ...
96B	Release ...
97A	Release ...

Use the Code List Editor to add, view and change information about a Code List. The Code List Editor is accessed through the Code Lists List window.

## Accessing Code Lists

- Validation Level
  - ▶ DT Rules
  - ▶ Send and Receive Usages



The Validation Level field on the Data Transformation rules and Send and Receive usages controls the level of automatic validation to be executed for EDI documents. Send and Receive mapping also allows you to use mapping commands on the literal or command line to control the validation level.

## Mapping Examples

### ■ Example 1

```

▶ If ( Validate ( "MYTABLE", \Table 1\20 M BEG\1 M 353\, False ) )
    mapping commands
Else
    Error ( 1, 5001, "Invalid value specified" )
EndIf

```

### ■ Example 2

```

▶ If ( Validate ( "MYTABLE", \Table 1\20 M BEG\1 M 353\, True ) )
    mapping commands
EndIf

```

The Data Transformation (DT) **Validate()** mapping function attempts to verify a character string by locating that value in a specified [Code List](#). The Validate() function uses the following format:

*result* **Validate** ( *table*, *charValue* [, *err* ] )

#### Where:

*table* – is a character string that identifies the name of the Code List. This is usually a [string constant](#), but can be a [variable](#), a [path](#) identifying a [simple element](#) in the source document definition, or an [expression](#) that evaluates to a character string.

*charValue* – is the source character string that will be looked up in the Code List. This is a variable, a path identifying a simple element in the source document definition, or an expression that evaluates to a character string.

*err* – is an optional flag indicating whether a not found condition will result in a warning message. If [True](#), a warning message will be issued if the value is not found in the Code List. If [False](#), no warning message will be issued. The default is **True**.

#### Returns:

*result* – is the Boolean value **True** or **False**. The value **True** is returned when the specified value is found in the Code List. **False** is returned when the specified value is not found in the Code List.

The *table* should exist as a valid Code List. If *table* does not exist, a warning message will be issued and it will be treated as a not found condition. The source character string is used to search the Code List. If the string is found in the Code List, **True** is returned by the function. If the string is not found in the Code List, **False** is returned.

# Data Format Example

The screenshot displays the WebSphere Data Interchange for Multiplatforms V3.3 - Development - Data Format Field - &FUNC\_ACK\_METADATA\_DICTIO... interface. The main window shows a list of fields in a dictionary, with the 'ACTION\_CODE' field selected. The detailed view of the 'ACTION\_CODE' field is shown in the foreground, including the following information:

Field Name	Dictionary Name	Description	Lock	Updated Date and Time	Updated User
ACCEPTED_MESSAGES	&FUNC_ACK_METADATA_DICTIO	Number of Accepted Messages	No	10/12/2006 2:49:32 PM	admin
ACTION_CODE	&FUNC_ACK_METADATA_DICTIO	Action Code (Accepted, Rejected, etc.)	Y...	10/12/2006 2:49:32 PM	admin
ASSOC_ASSIGNED_CODE	&FUNC_ACK_METADATA_DICTIO	Association Assigned Code	No	10/12/2006 2:49:32 PM	admin

The detailed view of the 'ACTION\_CODE' field includes the following fields:

- Field Name: ACTION\_CODE
- Dictionary Name: &FUNC\_ACK\_METADATA\_DICTIO
- Description: Action Code (Accepted, Rejected, etc.)
- Data Type: AN = Alphanumeric
- Field Length: 3
- Mapping Command: (empty)
- Literal: (empty)
- Send Map Validation:
- Receive Map Validation:
- Code List: (empty)
- Where Used: (empty)

The interface also shows a list of fields on the left side, including: ACCEPTED\_MESSAGES, ACTION\_CODE, ASSOC\_ASSIGNED\_CODE, CODE\_LIST\_VERSION, COMMON\_ACCESS\_REF, CONTROLLING\_AGENCY, DATA\_IMAGE, ELEMENT\_REF, ERR\_COMP\_ELEMENT\_F, ERR\_ELEMENT\_POS, ERR\_REP\_ELEMENT\_OC, FUNCTIONAL\_ID\_CODE, GROUP\_ACK\_CODE, GROUP\_CONTROL\_REF, IMPLEMENTATION\_AGE, IMPLEMENTATION\_REF, IMPLEMENTATION\_REL, IMPLEMENTATION\_VER, INTERCHANGE\_CONTR, LOOP\_ID, MESSAGE\_ACK\_CODE, MESSAGE\_CONTROL\_RI, MESSAGE\_RELEASE, MESSAGE\_SUB\_ID, MESSAGE\_TYPE, MESSAGE\_VERSION, MESSAGES\_INC, RECEIVER\_MESSAGES, RECEIVER\_ID, RECEIVER\_PARTNER\_ID.

At the bottom of the window, there is a footer with the text "WDI Code Lists" and "© 2007 IBM Corporation".

For the Send and Receive map processing, you can use the code list to validate a value by using an expression on the literal or command line. You can also enter a value into the Data Format field definition, Code List field to tell WDI to validate the contents of the field against values in the code list. This is not used for Data Transformation map processing.



## Restrictions

- Table value sizes are limited
- The maximum length of the code list entry can be from 1 to 35.
- The combined maximum length of the code list entry and its description cannot exceed 68.



The maximum length of the code list entry can be from 1 to 35 characters. The combined maximum length of the code list entry and its description cannot exceed 68 bytes.

## Internals

- DB2 Tables are EDIPSTD and EDIPSTV
  - ▶ EDIPSTD has table definitions for entry size and type
  - ▶ One row exists for each Table
  - ▶ Table name is the major key in both tables
  - ▶ Each code entry is a unique row in EDIPSTV
- Export / Import records are 8B1 and 8B2
  - ▶ 8B1 records contain EDIPSTD information
  - ▶ 8B2 records contain code values
    - VAR1 is the code value
    - VAR2 is a description, if any



There are two DB2 Tables in WDI that house Code Lists as well as Translation Tables. The EDIPSTD Table identifies each logical table. There is one row for each table. The key is the table name. Characteristics of the table, like size and type of the source element, size and type of the target element, are fields in the row. The EDIPSTV table contains the values of each table. Each value in the table is stored in its own row. Duplicate code list entries are not permitted (duplicate key) in a Code List.

The Export / Import subsystem creates a 8B1 record for each table based on the EDIPSTD table, and also one 8B2 record for each row in the EDIPSTV table.

## Summary

- Code Lists allow the user to validate the contents of a data element or field
- Implemented to allow validation of non-EDI fields
- Tables are named and can exist for every element
- Variable sizes can themselves vary by table
- Validate mapping command implements usage of a code list for DT maps
- Code lists are provided for most EDI elements when required by the Standards Body



In summary, Code Lists allow the user to validate the contents of an element or field. They are implemented to allow validation of non-EDI fields in validation maps. EDI fields are validated automatically during syntax validation. A Code List can exist for a number of elements. Code Lists are named and can be referenced in mapping commands. Each entry for a code list is the same size, but each code list can have different sizes for its primary element. The Validate mapping command implements the ability to use a translation table for DT maps.

# 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	WMO	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
ef (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.