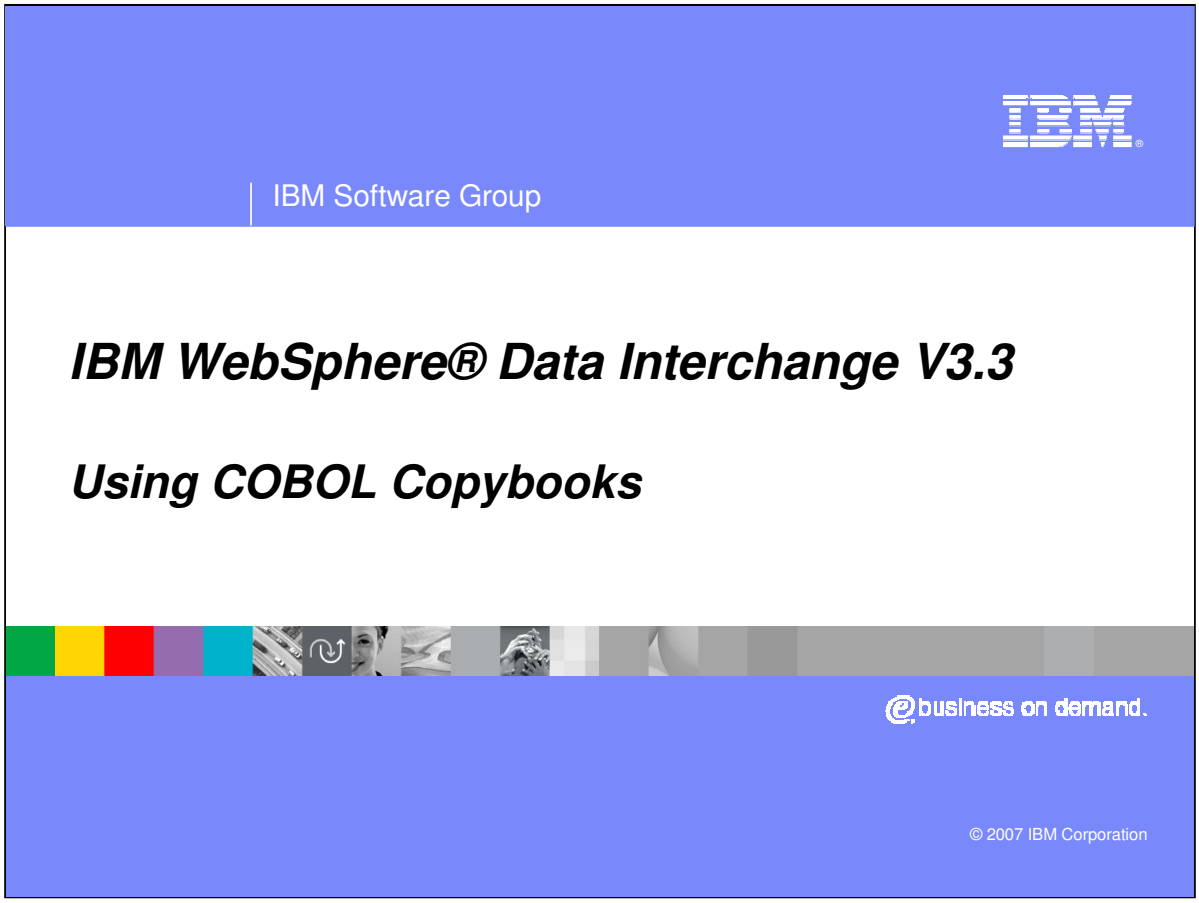


This presentation will review using COBOL Copybooks with Data Format Definitions.

Agenda

- Review Data Formats
- Describe how to use COBOL Copybooks
- COBOL Copybook example

The presentation will review the steps for using COBOL Copybooks.

Using COBOL Copybooks

- Defines the layout of your application data
- Records, field names, and lengths
- Repeating structures and loops

The term *data format* defines the layout of your application data. It is a document or metadata definition. The word *data* refers to the information itself. The word *format* refers to the physical layout of information in the file, such as field names and lengths. WebSphere Data Interchange requires a description of the data format for each business application that generates data for translation, or uses translated data. Application data must be described to WebSphere Data Interchange so that it can be used as either a source or target for translation.

Using COBOL Copybooks

- Defining Data Formats
- Create Data Format Dictionary
  - Create Data Format Definitions or Document
    - Create Record definitions
      - Create Structures
        - Create Fields
      - Create Fields
    - Create Loops
  - Import Cobol Copybook
    - Imports records, structures, fields, creates code lists

The concepts for defining a data format are similar to Electronic Data Interchange (EDI) Standards. You Define a Data Format Dictionary. The Dictionary contains components for field, structure, record, and loop definitions. The data format definition contains record and loop definitions for the business document layout. Records can contain structures which contain fields and fields. All the components within a Data Format Dictionary can be copied, updated, and deleted and all components can be re-used in different business document definitions. For example a record can be used in 2 different data format definitions.

COBOL copybooks can be imported into a Data Format Dictionary. You can use this mechanism to create or update Data Format Record, structures, fields, and code lists. The imported Records, Structures, and Fields will be a part of the Data Format Dictionary into which they are imported. The Data Format Records can be used in a existing Data Format or a new Data Format.

Using COBOL Copybooks

```
01  COBOL-TEST-RECORD.
      05  COBOL-TEST-USAGES.
        10  COBOL-4-COMP      PIC S9(4)  COMP.
        10  COBOL-8-COMP      PIC S9(8)  COMP.
        10  COBOL-9-COMP      PIC S9(9)  COMP.
        10  COBOL-4-COMP3     PIC S9(4)  COMP-3.
        10  COBOL-5-COMP3     PIC S9(5)  COMP-3.
        10  COBOL-6-COMP3     PIC S9(6)  COMP-3.
        10  COBOL-7-COMP3     PIC  9(7)  COMP-3.
        10  COBOL-4-COMP2     PIC S9(4)  COMP-2.
        10  COBOL-5-COMP2     PIC S9(5)  COMP-2.
        10  COBOL-6-COMP2     PIC S9(6)  COMP-2.
        10  COBOL-7-COMP2     PIC  9(7)  COMP-2.
```


This is an example of a COBOL copybook.

Using COBOL Copybooks

```
05  COBOL-TEST-OCCURS.
        10  COBOL-1-OCCURS     PIC S9(4)  COMP
                               OCCURS 10 TIMES
                               INDEXED BY idx1.
        10  COBOL-2-OCCURS     PIC S9(5)  COMP-3
                               OCCURS 8
                               INDEXED BY idx1 idx2.
        10  COBOL-3-OCCURS-group
                               OCCURS 5.
          15  COBOL-3-OCCURS PIC S9(4)  COMP.
          15  COBOL-4-OCCURS PIC X(8).
```


This is an example of a repeating structure within a record.

Using COBOL Copybooks

```
05  COBOL-TEST-REDEFINES.
        10  COBOL-redef-area.
          15  COBOL-1-area   PIC X(10).
          15  COBOL-2-area   PIC X(10).
          15  COBOL-3-area   PIC X(5).
        10  COBOL-redef-1 REDEFINES COBOL-redef-area
                    PIC X(25).
        10  COBOL-redef-2 REDEFINES COBOL-redef-area.
          15  COBOL-a-area   PIC X(15).
          15  COBOL-b-area.
            20  COBOL-b1-area   PIC X(7).
            20  COBOL-b2-area   PIC X(3).
        10  COBOL-redef-3 REDEFINES COBOL-redef-2
                    PIC X(15).
```

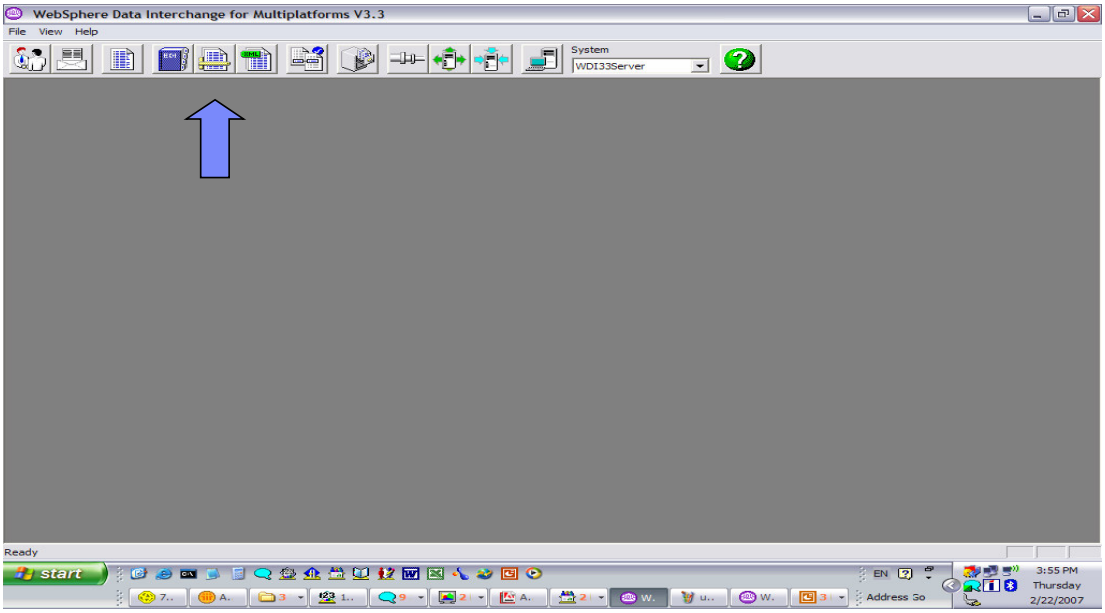

This is an example of a structure re-definition within a record.

Using COBOL Copybooks

```
05  COBOL-TEST-features.
        10  COBOL-feat-1       PIC S(4)  COMP  SYNC.
          88  COBOL-feat-1-a  VALUE 17  22.
          88  COBOL-feat-1-b  VALUE +12.
          88  COBOL-feat-1-c  VALUE -3.3.
        10  COBOL-FEAT-2       PIC S999V99  COMP-3.
          88  COBOL-FEAT-2-A  VALUE 10.20.
          88  COBOL-FEAT-2-B  VALUE +8.13.
          88  COBOL-FEAT-2-C  VALUE +5.05.
          88  COBOL-FEAT-2-D  VALUE 2 THRU 7.
```

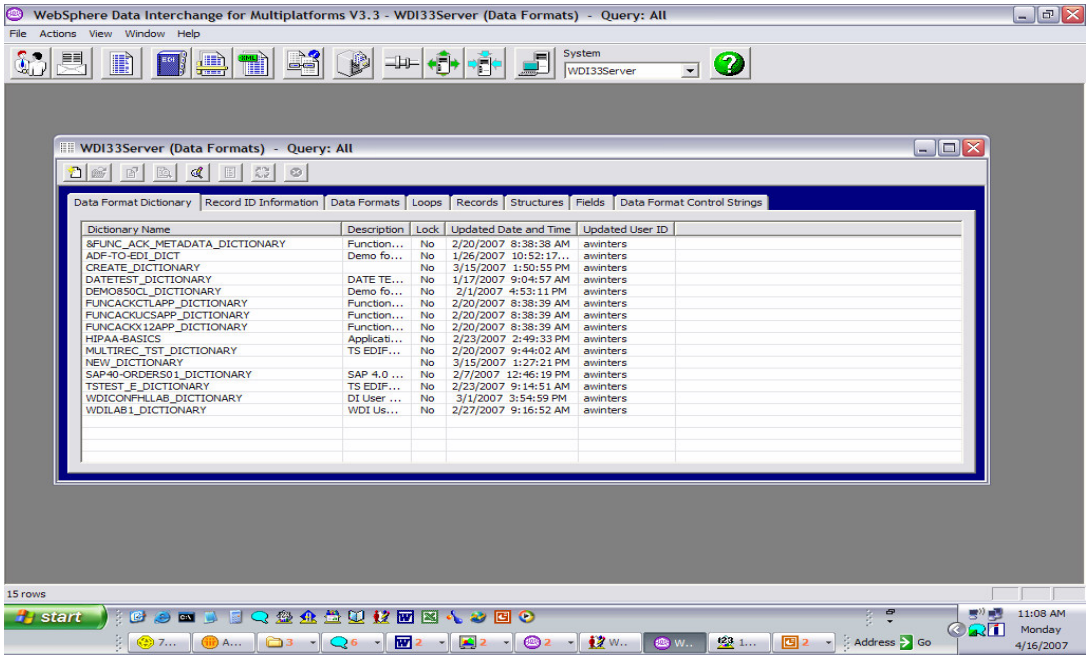

This is an example of a level 88 to be used for creating code lists.

Data Formats



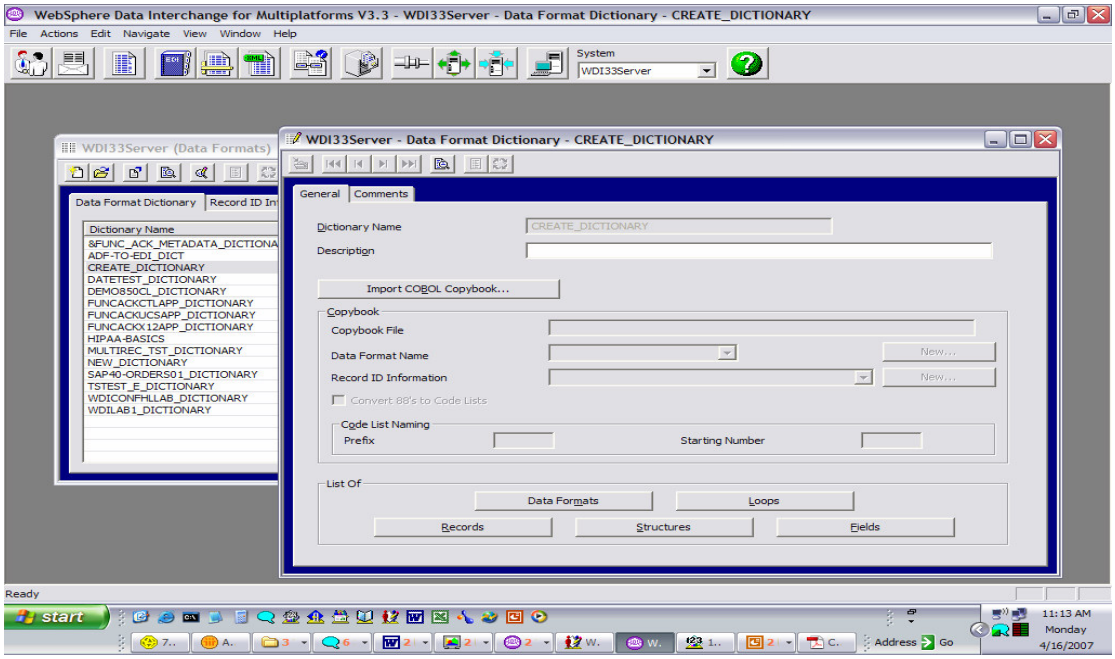
This is the location of the WebSphere Data Interchange (WDI) Client Data Formats Functional Area.

Data Formats



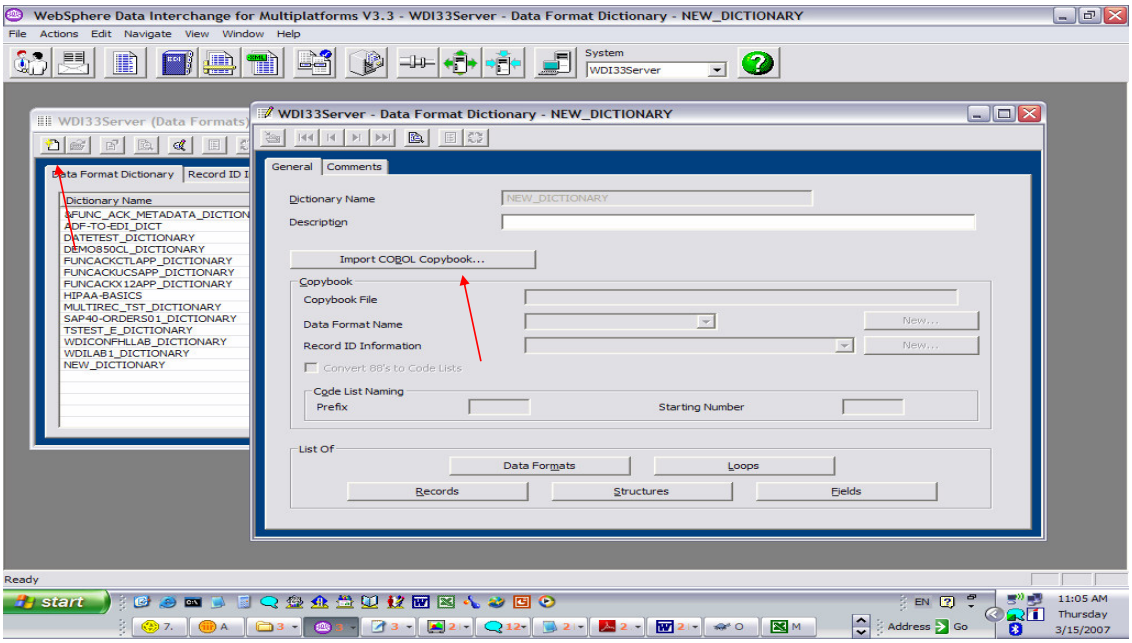
Components of Data formats are dictionary, record ID information, data formats, loops, records, structures, and fields. These are similar to Electronic Data Interchange (EDI) Standards components but describe your application data. The dictionary contains the component definitions for fields, structures, records, loops which allows you to re-use components within different data format definitions. Data format definitions contain records and loops. Loops contain records. Records contain fields and structures. And structures contain fields.

Data Formats



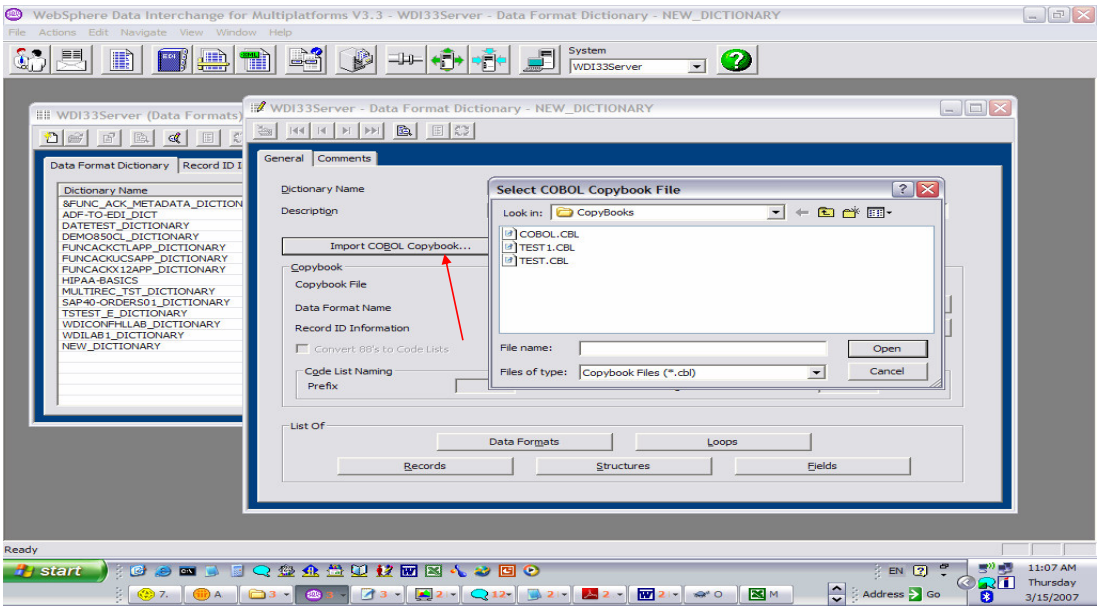
All components except for record ID information are available when selecting a specific Data Format Dictionary.

Using COBOL Copybooks



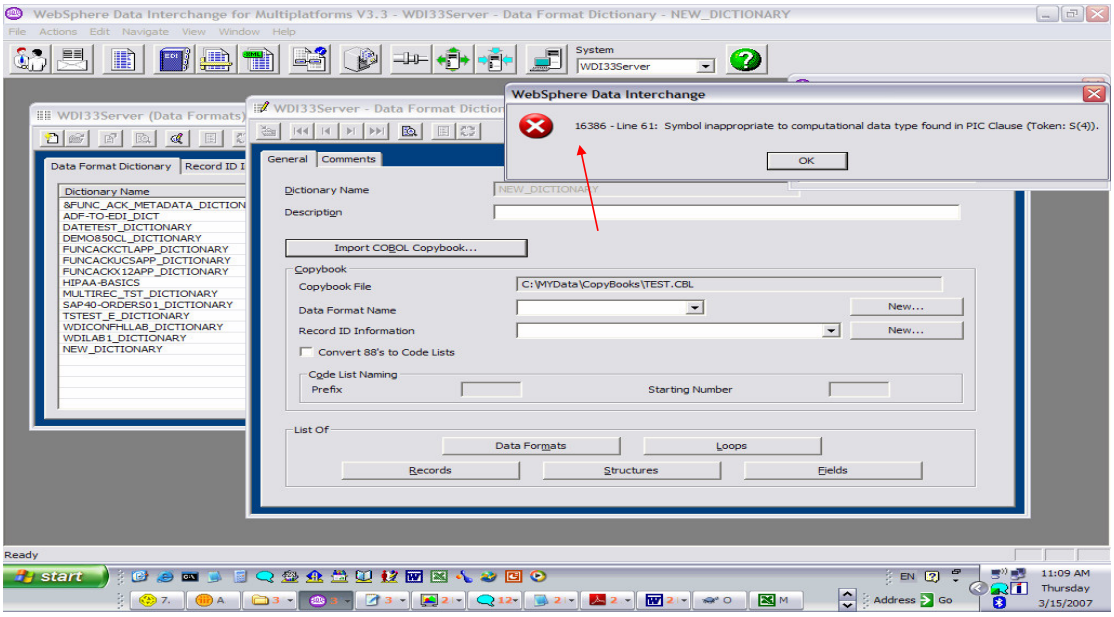
The first step in using a copybook is to create or select a data format dictionary to hold the definitions in the copybook. With this example we will create a new Dictionary. Enter the Dictionary Name and save to activate the Import COBOL Copybook button. Next select the Import COBOL Copybook button.

Using COBOL Copybooks



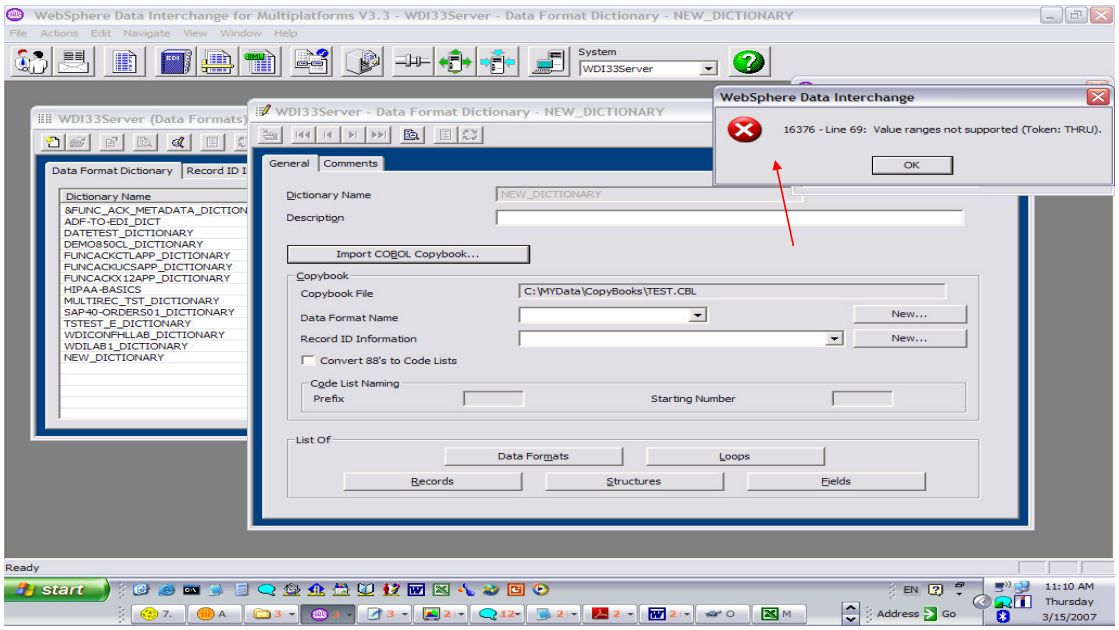
You can navigate to the location of the copybook. When you make your selection and press Open the import will begin.

Using COBOL Copybooks



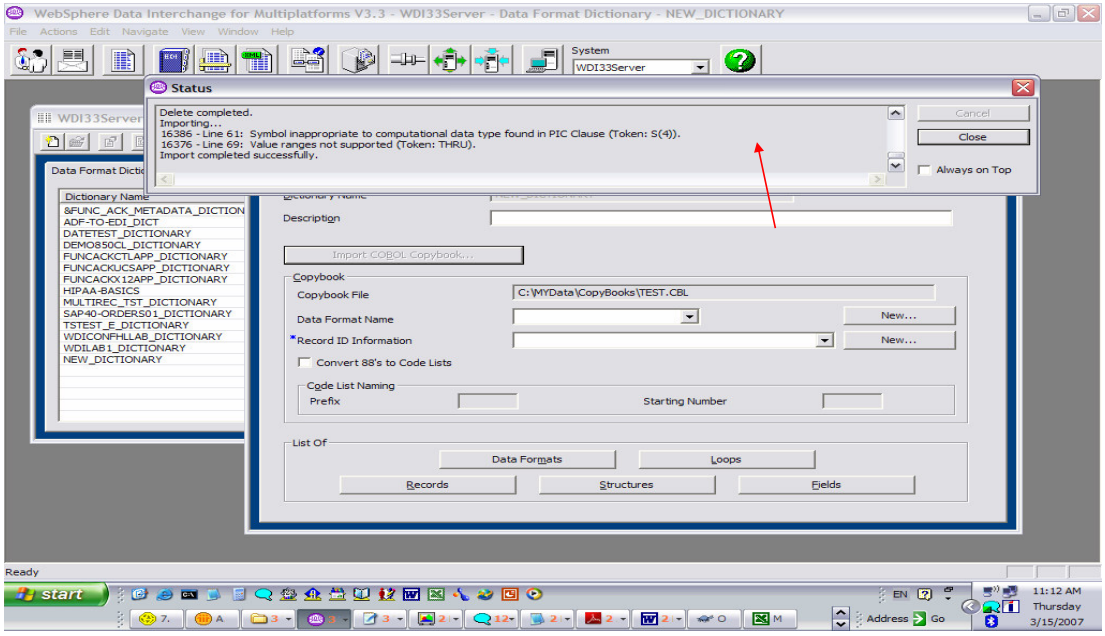
With this example looks like there could be some issues with the copybook. For now we will continue the import by pressing OK.

Using COBOL Copybooks



More issues with our example, we will continue the import by pressing OK.

Using COBOL Copybooks



When the import has completed we will get a Status window which also contains the issues found. Lets look at the first issue.

Using COBOL Copybooks

16386 - Line 61: Symbol inappropriate to computational data type found in PIC Clause (Token: S(4)).

```
05  COBOL-TEST-features.
        10  COBOL-feat-1      PIC S(4)  COMP  SYNC.
          88  COBOL-feat-1-a  VALUE 17  22.
          88  COBOL-feat-1-b  VALUE +12.
          88  COBOL-feat-1-c  VALUE -3.3.
```

```
10  COBOL-feat-1      PIC S9(4)  COMP  SYNC.
```


The first issue is 16386. If you look at the copybook file, the PIC S(4) is incorrect and should be PIC S9(4)

Using COBOL Copybooks

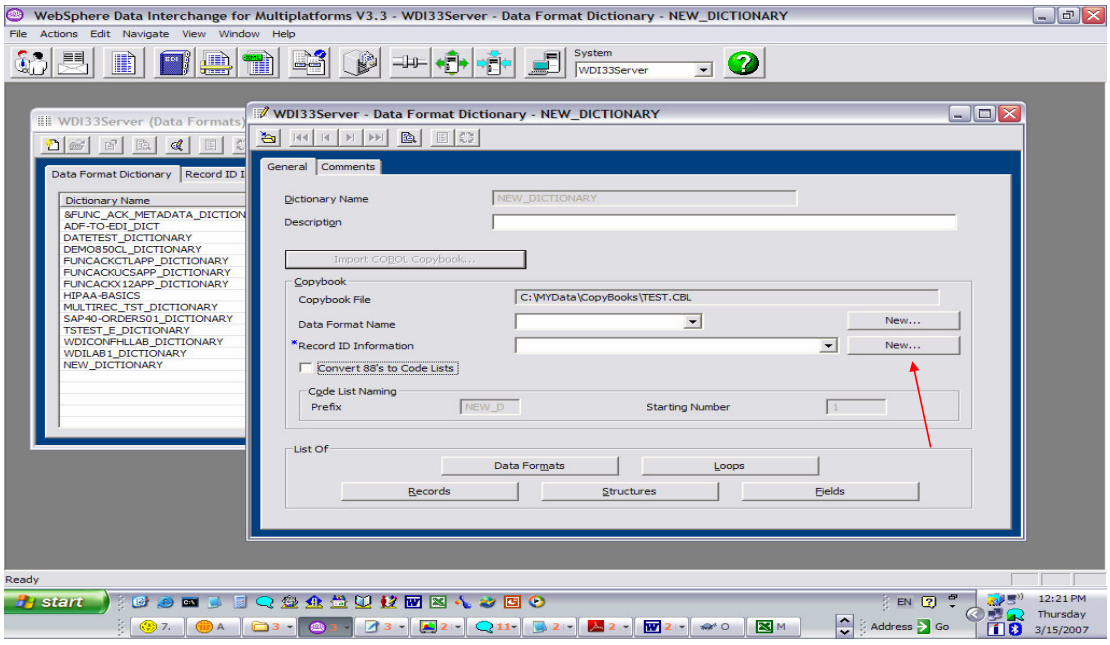
16376 - Line 69: Value ranges not supported (Token: THRU).

```
05  COBOL-TEST-features.
        10  COBOL-feat-1      PIC S9(4)  COMP  SYNC.
          88  COBOL-feat-1-a  VALUE 17  22.
          88  COBOL-feat-1-b  VALUE +12.
          88  COBOL-feat-1-c  VALUE -3.3.
        10  COBOL-FEAT-2      PIC S999V99  COMP-3.
          88  COBOL-FEAT-2-A  VALUE 10.20.
          88  COBOL-FEAT-2-B  VALUE +8.13.
          88  COBOL-FEAT-2-C  VALUE +5.05.
          88  COBOL-FEAT-2-D  VALUE 2 THRU 7.
```

```
          88  COBOL-FEAT-2-D  VALUE 2.
          88  COBOL-FEAT-2-D  VALUE 3.
          88  COBOL-FEAT-2-D  VALUE 4.
          88  COBOL-FEAT-2-D  VALUE 5.
          88  COBOL-FEAT-2-D  VALUE 6.
          88  COBOL-FEAT-2-D  VALUE 7.
```

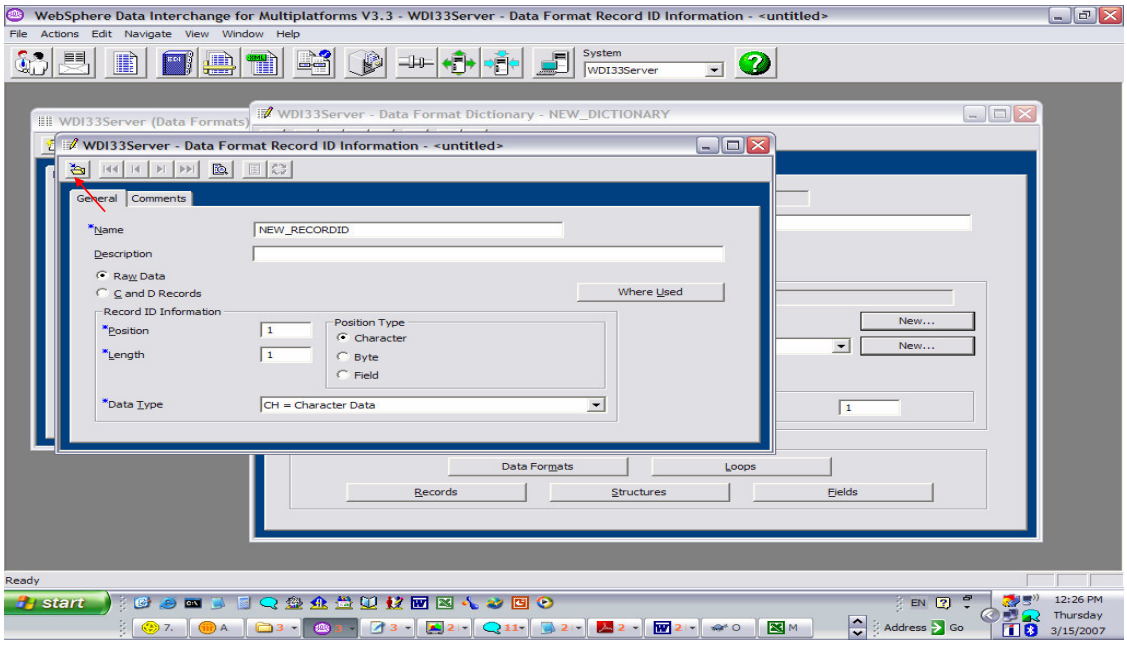

The next issue is 16376. Value ranges are not supported. You can correct this by changing the level 88 with the range identification. Each value can have a level 88 with the values.

Using COBOL Copybooks



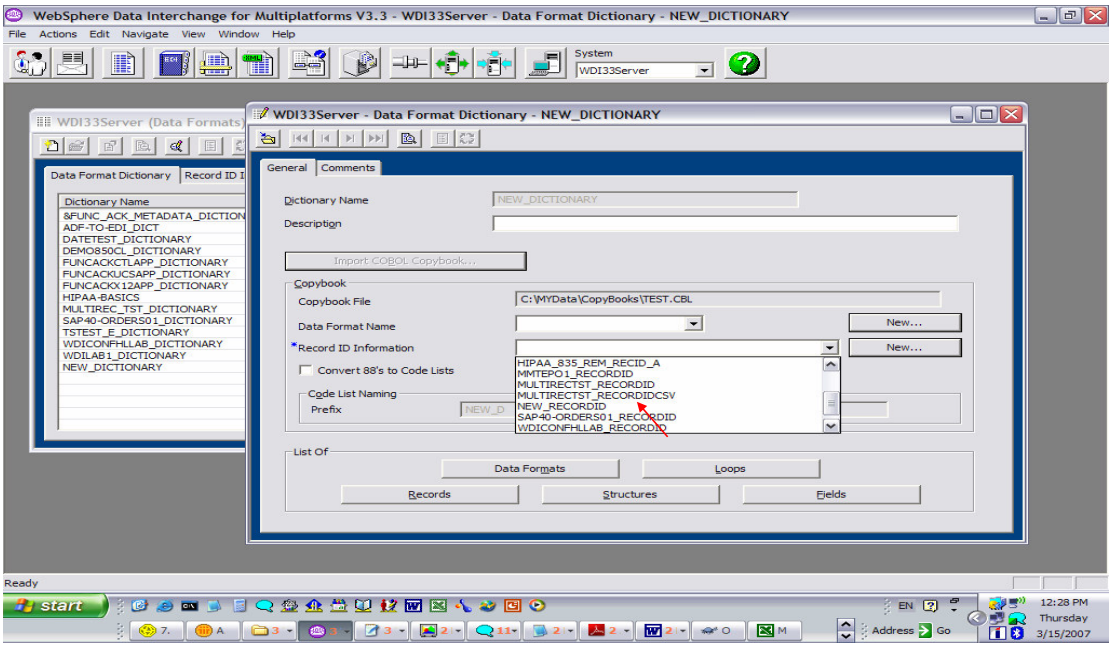
The next step is to select a record ID information object. You can also create a new one.

Using COBOL Copybooks



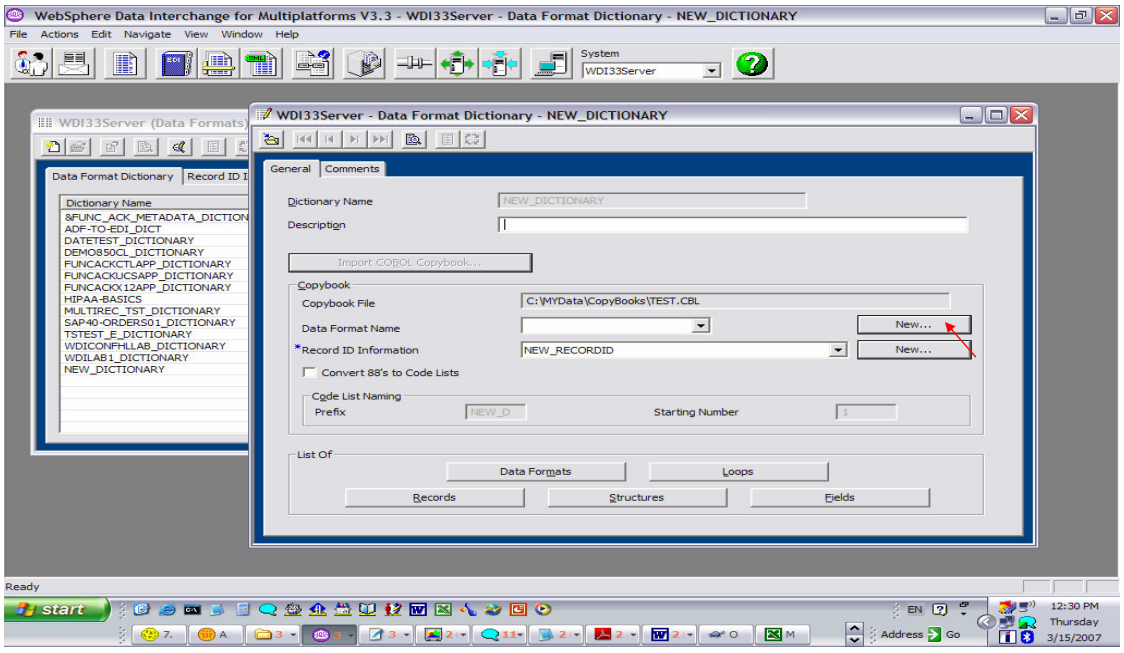
With a new Record ID Information object, define the record ID Information and press the save button.

Using COBOL Copybooks



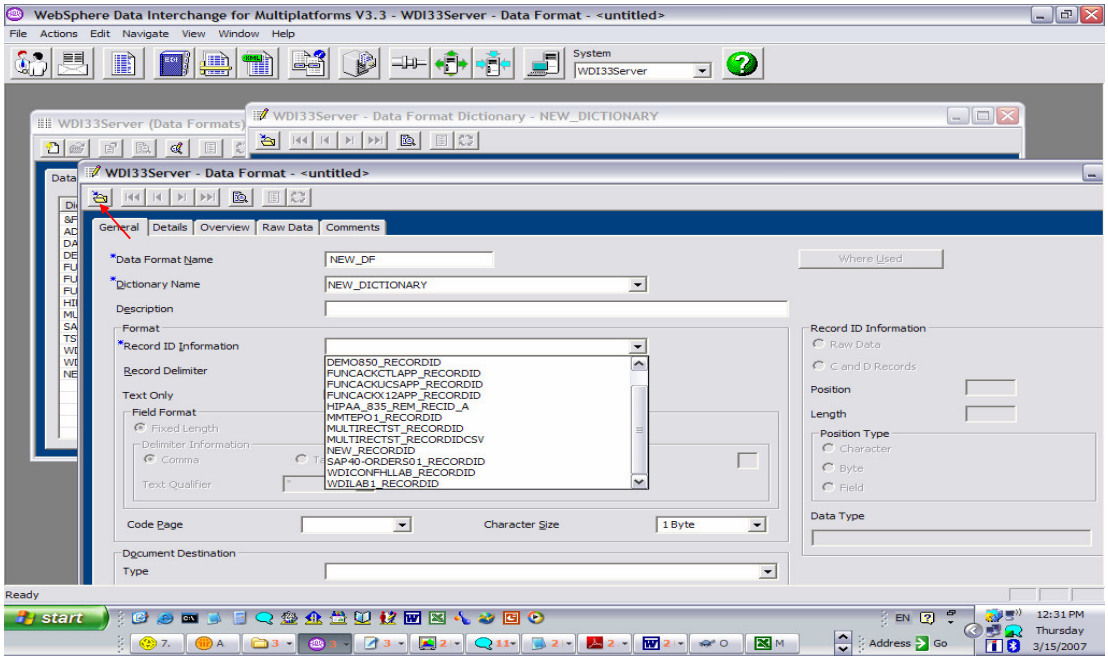
Select the Record ID Information object.

Using COBOL Copybooks



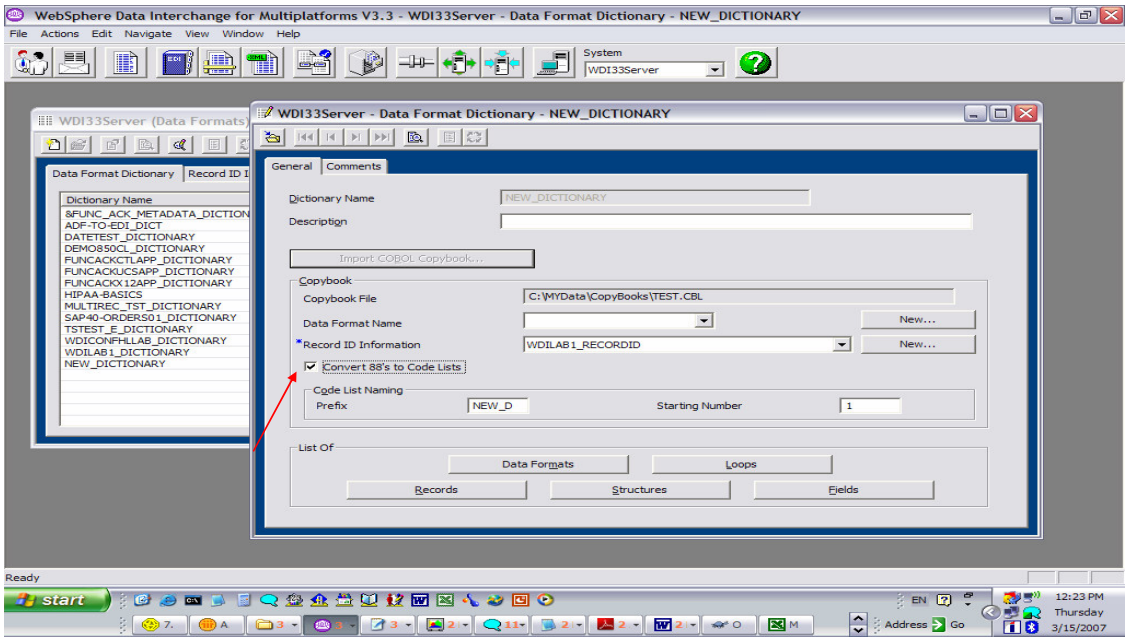
You can also select or create a Data Format Definition.

Using COBOL Copybooks



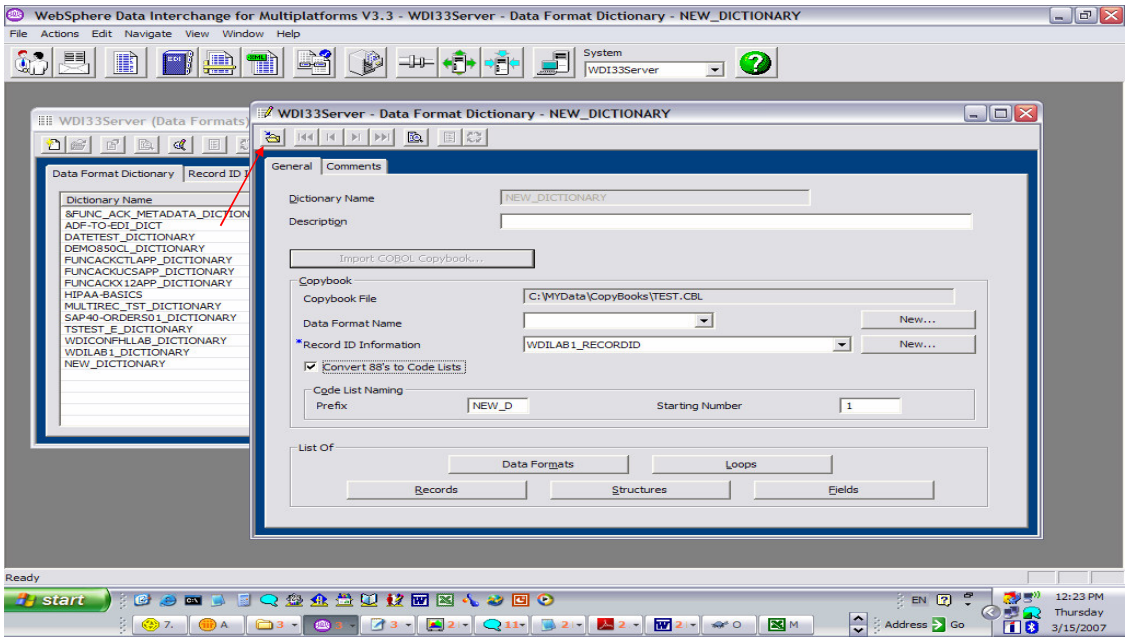
Complete the required information and press the Save button.

Using COBOL Copybooks



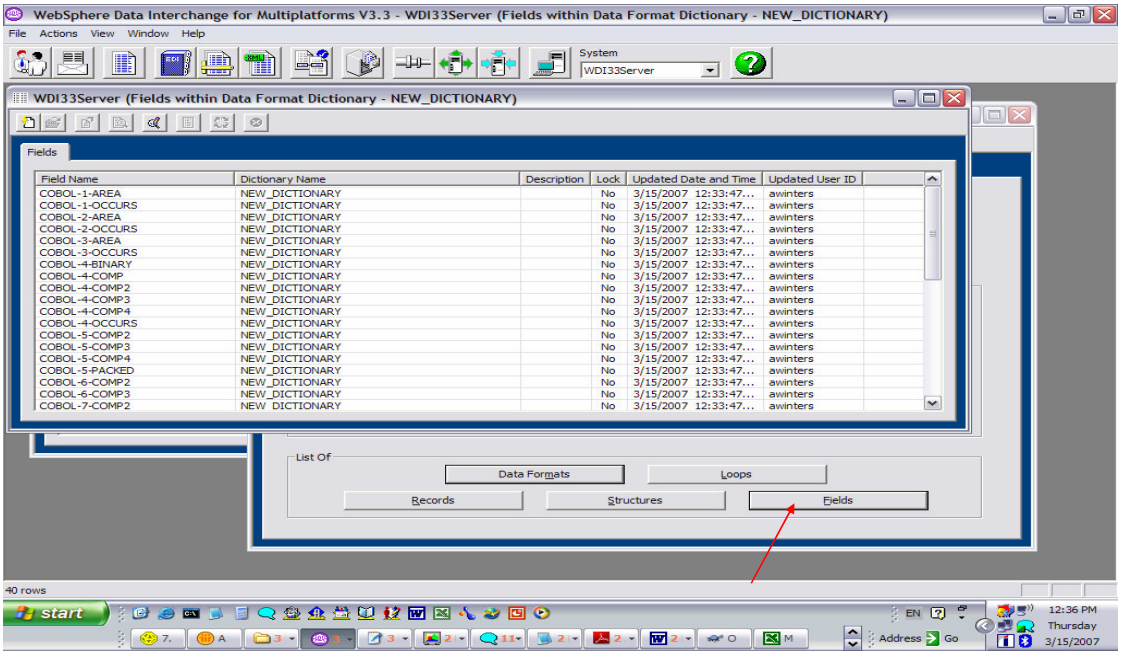
The next step is to select for level 88 conversions to WebSphere Data Interchange (WDI) code lists. The names assigned to the Code List are determined by the values in the Prefix and Starting Number fields. Prefix is a value up to five characters in length that will be used as the first part of the name for all created Code Lists. The default is the first 5 characters of the name of the Data Format Dictionary. The Starting Number field is a one to three digit number. The first Code List created by importing the COBOL copybook will have this value appended to the specified prefix to form the name of the Code List. WebSphere Data Interchange Client will increment this value for each subsequent Code List created and use the incremented value and the prefix as the name of the Code List.

Using COBOL Copybooks



Now you can press the Save button. Lets see what happened.

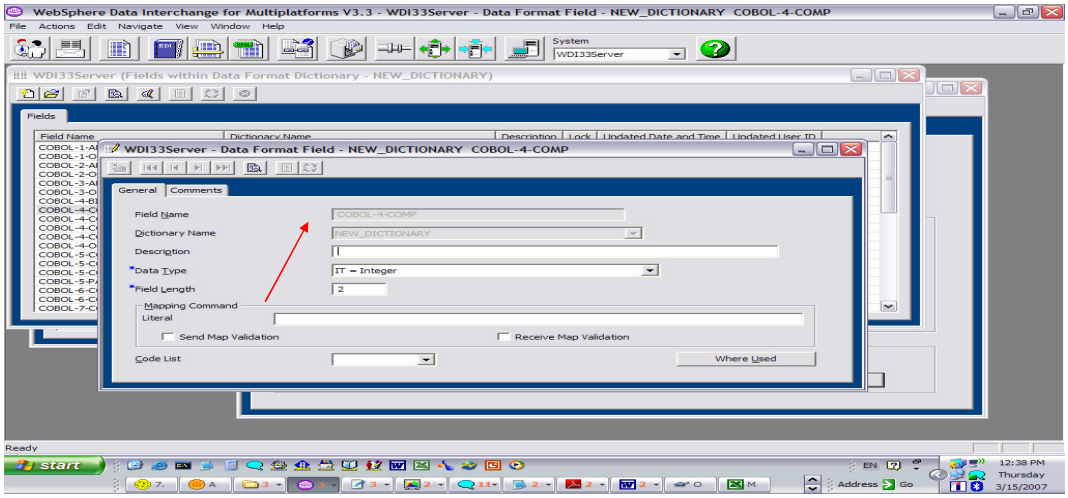
Using COBOL Copybooks



This is a list of the fields that were imported.

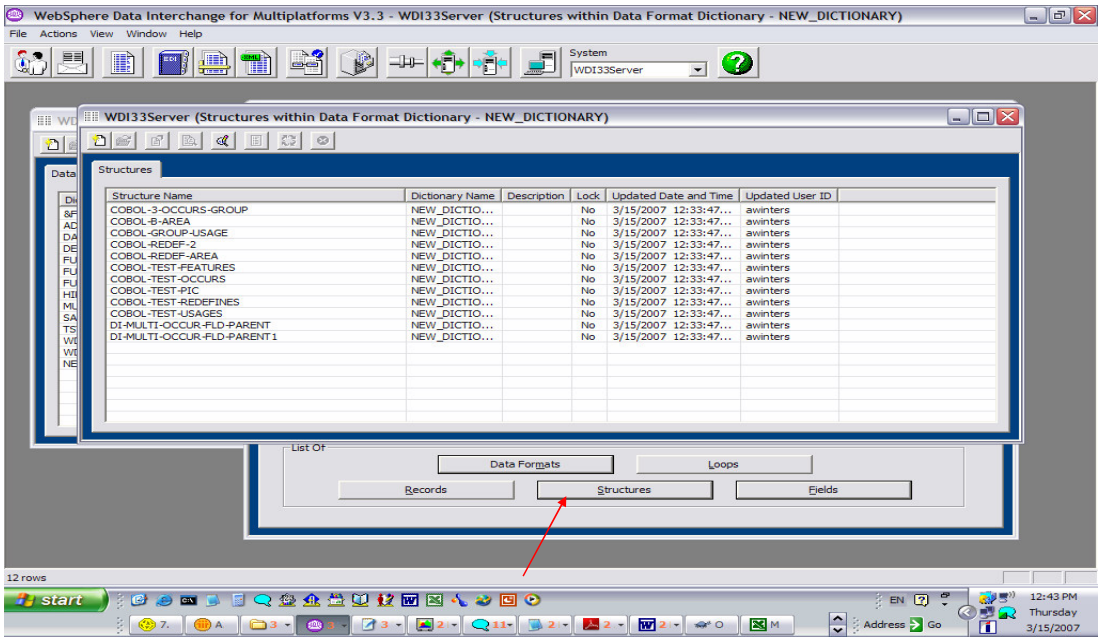
Using COBOL Copybooks

```
01  COBOL-TEST-RECORD.
      05  COBOL-TEST-USAGES.
        10  COBOL-4-COMP       PIC S9(4)  COMP.
```



You can open the field and check or modify the definition.

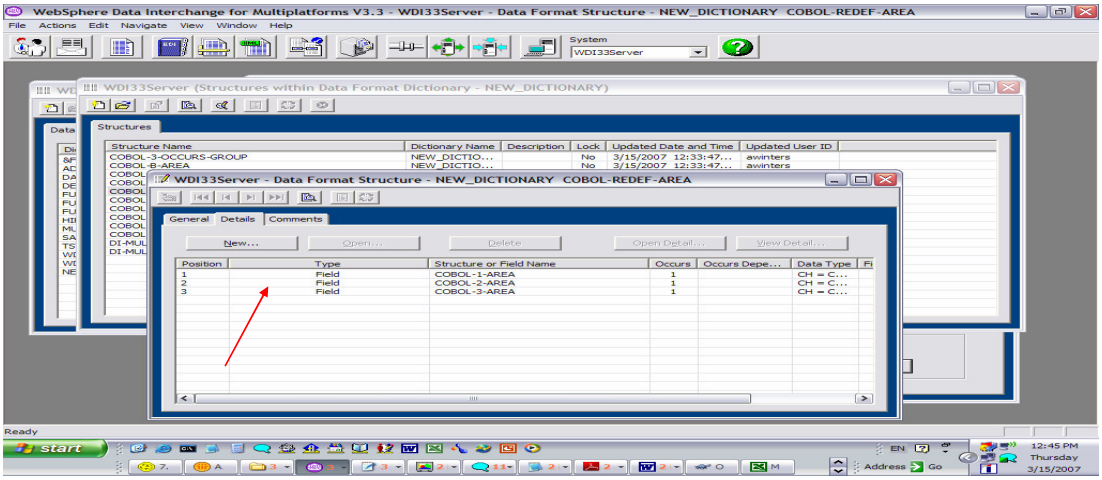
Using COBOL Copybooks



This is a list of the structures that were imported.

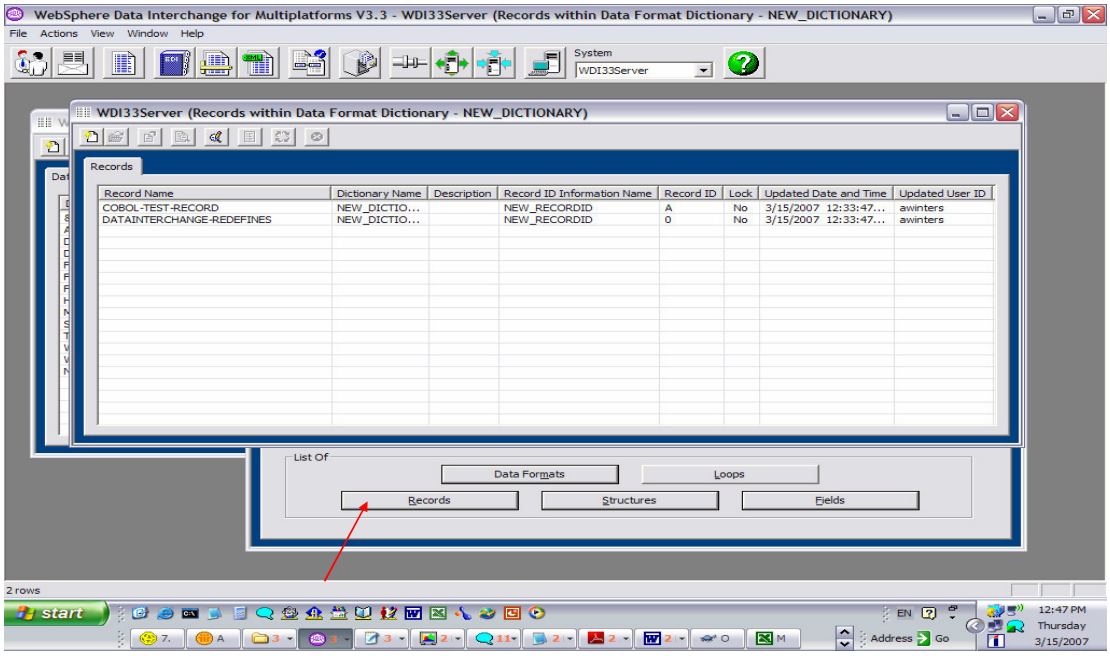
Using COBOL Copybooks

```
05  COBOL-TEST-REDEFINES.
        10  COBOL-redef-area.
            15  COBOL-1-area   PIC X(10).
            15  COBOL-2-area   PIC X(10).
            15  COBOL-3-area   PIC X(5).
```



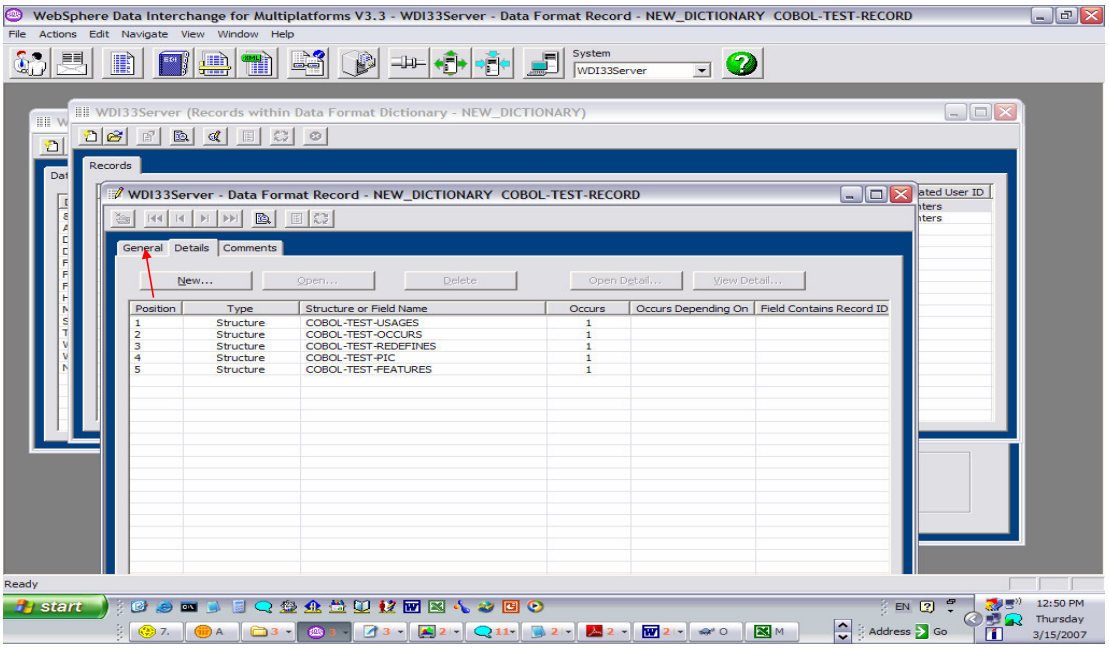
You can open the structure and check or modify the definition.

Using COBOL Copybooks



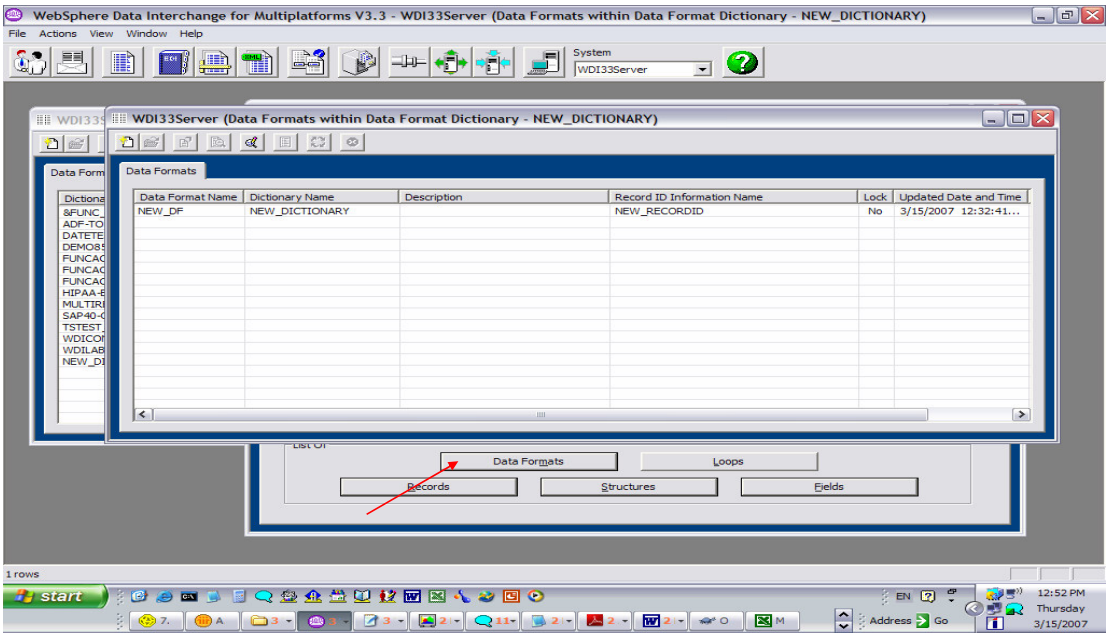
This is a list of the records that were imported.

Using COBOL Copybooks



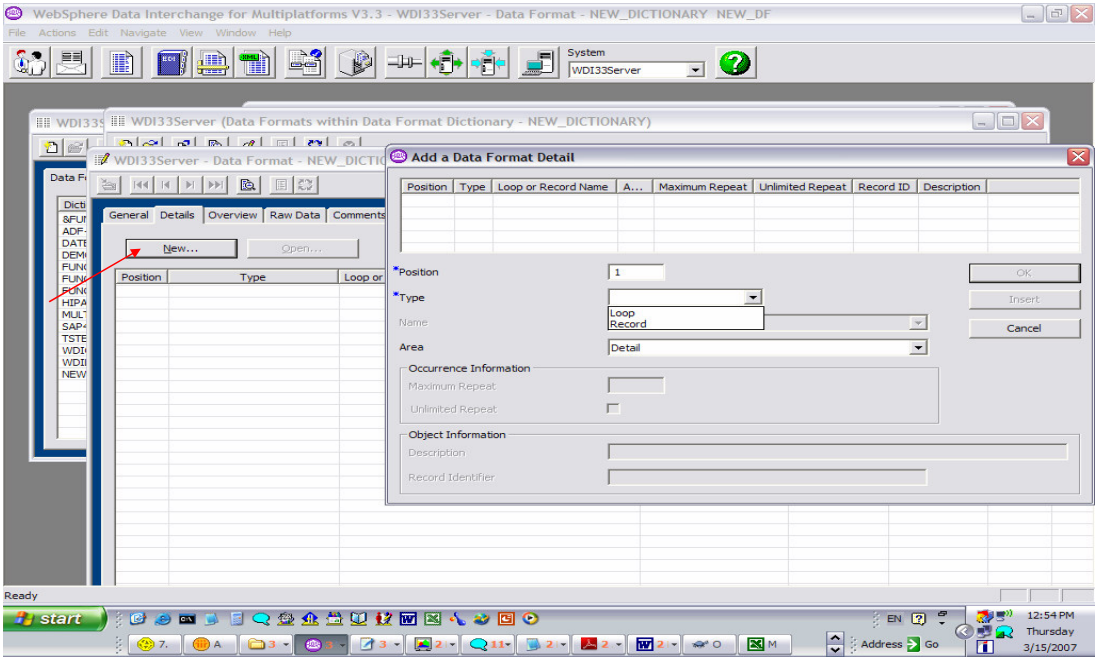
You can modify the record as needed. The Record ID value is located on the General Tab.

Using COBOL Copybooks



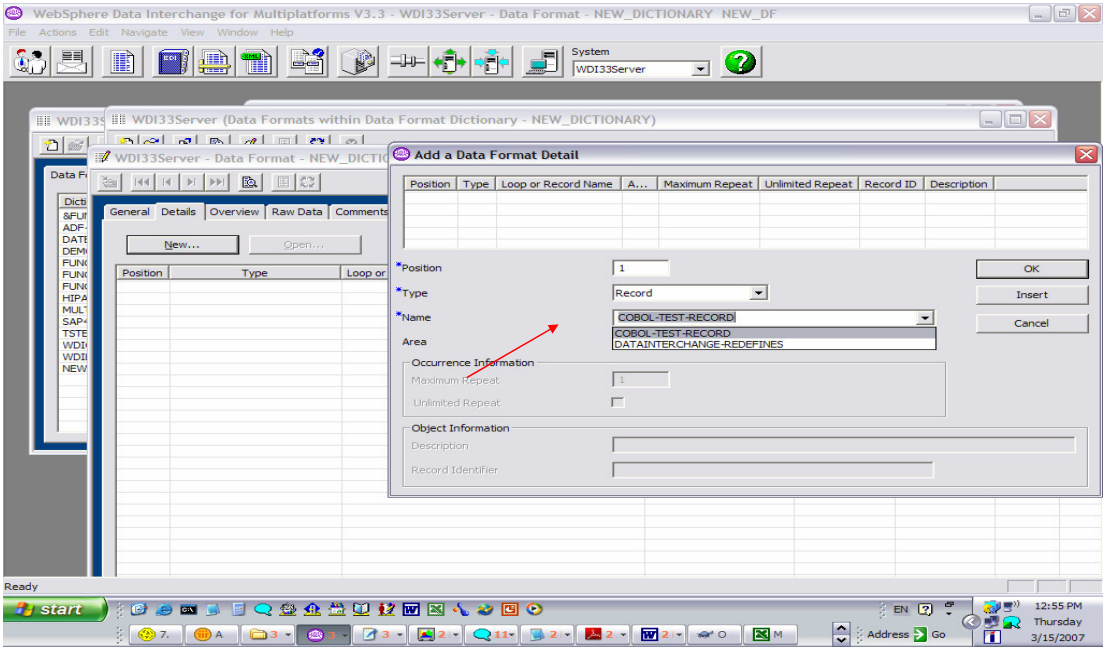
This is a list of the Data formats.

Using COBOL Copybooks



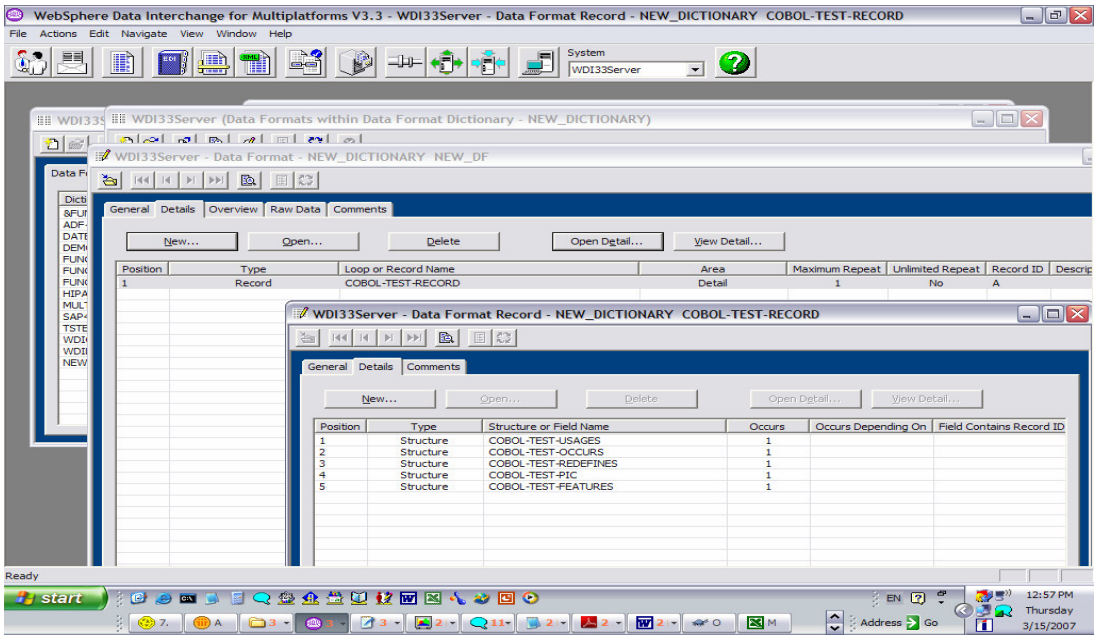
Initially the Data Format does not contain any details. Select the New button to begin defining the Data Format. Select Type Loop or Record.

Using COBOL Copybooks



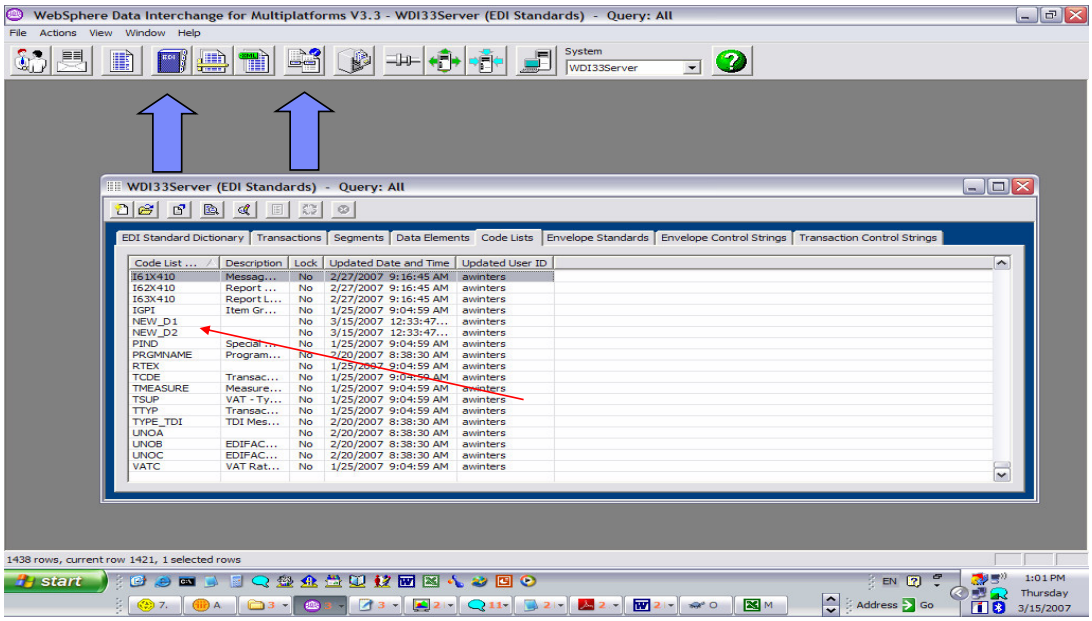
And select the Record or Loop name from the Drop Down list.

Using COBOL Copybooks



Now you have a Detail record. All the components for the record are included in the Data Format definition. You can select and modify the components using the Data Format editors at this location or on the Data Format Dictionary General Tab.

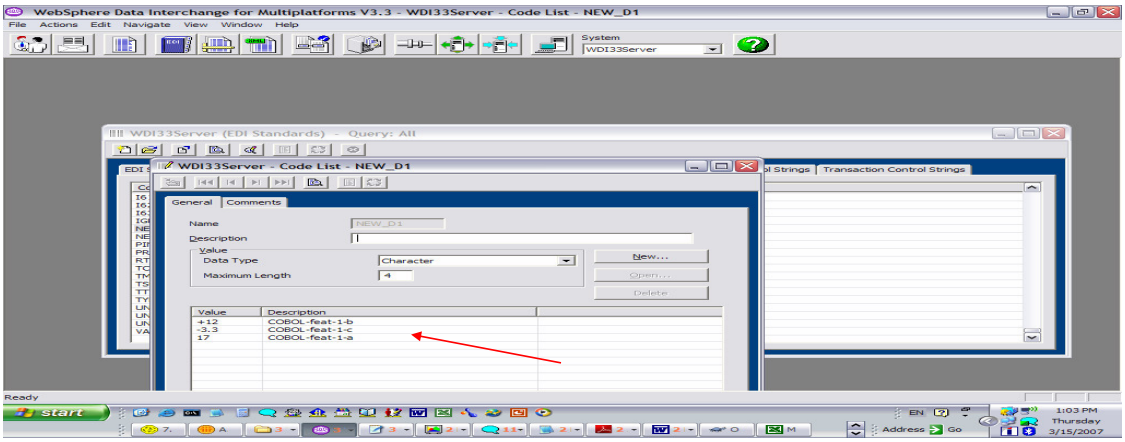
Using COBOL Copybooks



The level 88s were converted to code lists. The code lists are located in the EDI Standard and Mapping functional areas. These are the code lists associated with the level 88s in the example COBOL Copybook file imported.

Using COBOL Copybooks

```
05  COBOL-TEST-features.
        10  COBOL-feat-1      PIC S9(4)  COMP  SYNC.
            88  COBOL-feat-1-a  VALUE 17  22.
            88  COBOL-feat-1-b  VALUE +12.
            88  COBOL-feat-1-c  VALUE -3.3.
```



The level 88s were converted to code lists. These are the values from the level 88s

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.

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.