

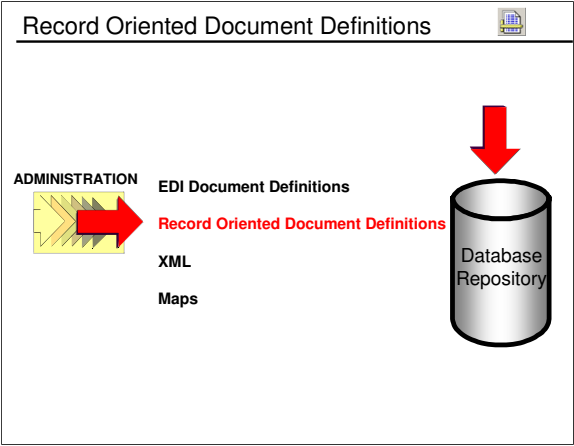
Define the layout of Record Oriented data.

## Unit Objectives

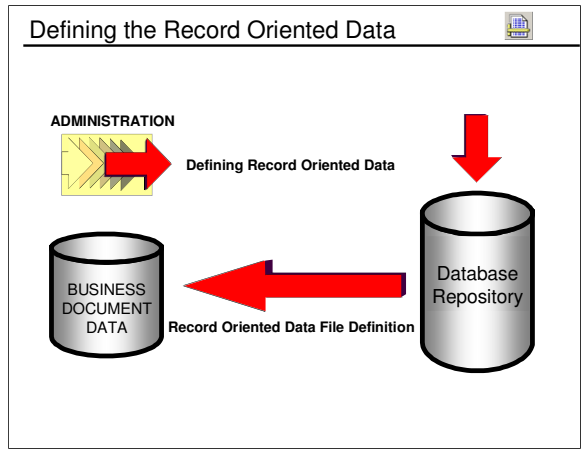


- Define Record Oriented Data (ROD)
- Create a ROD dictionary
- Define Record ID Information Objects
- Define ROD documents
- Define Loops and repeating records
- Define Records
- Define Structures
- Define Fields and specify field attributes

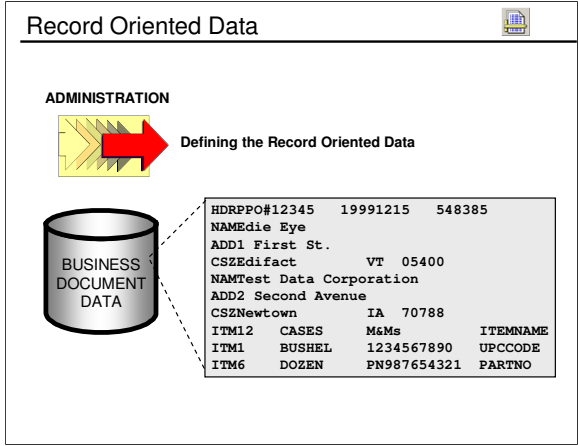
Each of these objectives are addressed in this module.



Record Oriented Data objects are defined in Data Interchange Services and placed in the selected database.



As an EDI document definition defines the layout of an EDI document and an XML DTD or Schema defines the layout of an XML document, Record Oriented Document Definitions are required to define the layout of fixed-length field and comma-separated data.



Record Oriented Data must be a flat file with fixed-length fields or comma separated values.

### Two Application Data Options



- C&D Records
  - Sequential data set (flat file)
  - The first record is a control (C) record
  - Remaining records are (D) records
- Raw Data
  - Sequential data set (flat file)
  - Each record format is identified with a unique record ID
  - Data identifying the trading partner must be in one of the records
  - Either the beginning or ending record type may not repeat
  - Data may consist of fixed length fields or Comma Separated Values (CSV)


The original format for application data was C&D records. Most implementations today use raw data as it is a more flexible format. C&D record formats are the original format for application data for the DataInterchange product. They are maintained for backward compatibility.

### Selecting C&D or Raw Data



- Advantages of C&D Records
  - Multiple Record Oriented Document Definitions may be contained in a single file
  - Envelope overrides may be specified in the C record
- Advantages of Raw Data
  - Application output may be used directly or with a minimum amount of modification
  - Comma-separated values are allowed

Some companies prefer C&D records as the C record identifies the Record Oriented Document Definition, allowing mixed document types to be processed within a single input file.

**C&D Records** 

- C Record
  - Character "C" in column 1
  - Columns 2-36 – Trading Partner identifier
  - Columns 37-52 – Record Oriented Document Definition name
  - ...etc., including: Return codes, Test/Prod indicator, Sender/Receiver IDs, Group envelope values, and other parameters
- D Record
  - Character "D" in column 1
  - Record name in columns 2-17
  - Data fields in column 18+
- Z Record
  - End of data indicator

C and D record formats are pre-defined by IBM. The “D” records or data records use columns 1 through 17 as the record ID. With raw data, you define the position and length of the record ID field. Chapter 3 of the Programmer’s Reference provides full details regarding the layout of C and D records.




### Record Oriented Data Requirements




- Sequential data set (flat file)
- Each record format is identified with a unique record ID
- Data identifying the trading partner must be in one of the records
- Either the beginning or ending record type may not repeat
- Data may consist of fixed length fields or Comma Separated Values (CSV)

Record Oriented Data must conform to the criteria identified in the slide.

Loops, Records, and Structures 

RECORD 1 -	HDRPO#12345	19991215	548385		
LOOP1 {	RECORD 2 -	NAMEdie Eye			
	RECORD 3 -	ADD1 First St.			
	RECORD 4 -	CSZEdifact	VT	05400	
LOOP1 {	RECORD 2 -	NAMTest Data Corporation			
	RECORD 3 -	ADD2 Second Avenue			
	RECORD 4 -	CSZNewtown	IA	70788	
	RECORD 5 -	ITM12	CASES	M&Ms	ITEMNAME
	RECORD 5 -	ITM1	BUSHEL	1234567890	UPCCODE
	RECORD 5 -	ITM6	DOZEN	PN987654321	PARTNO

 STRUCTURE

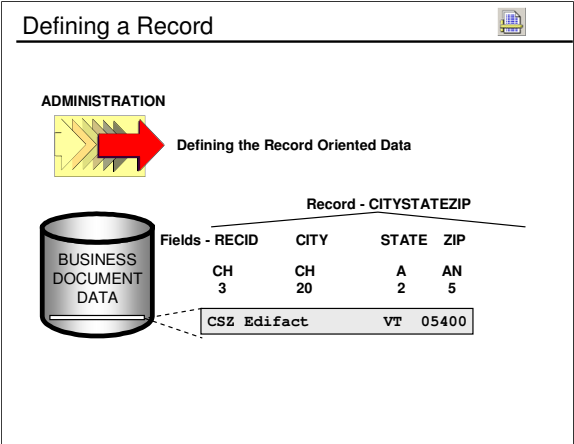
Record Oriented data allows you to define the positions of the record IDs and fields. The PERFORM statement provides the name of the Record Oriented Document definition. The keyword used in the TRANSFORM command is “DOCUMENT”.

### **Dictionaries and Record IDs**



- **ROD Dictionaries provide a repository for fields, structures, records, and loops that are defined by Record Oriented Document Definition**
- **Each dictionary can contain multiple Record Oriented Document Definitions that share fields, structures, records, and loops**
- **Record ID Information is required for Record Oriented Data**
- **Record IDs are defined by data type, position in the record and length**
- **Record ID Information may be used by multiple Record Oriented Document Definitions**
- **Each Record Oriented Document Definition is associated with a dictionary**
- **Record Oriented Document Definitions may use the same field name (as well as loop, record, or structure name) multiple times**

A Record Oriented Data Dictionary essentially is a name within which other components are grouped. A Record Oriented Data Dictionary allows you to build a library of Record Oriented Data building blocks for specific Record Oriented Document Definitions. For instance, you could create a Record Oriented Document Dictionary for your purchasing system with the idea that many purchasing Record Oriented Document Definitions would use the same components, all of which could be stored in a single Record Oriented Document Dictionary. Those components include Loop, Record, Structure, and Field definitions.



Each record is named in the Record Oriented Document definition. Records contain field definitions and may contain structures identifying groups of fields. Each field is identified as to data type and length. Data Interchange Services calculates the offset position of each field. Therefore if there is unused space, dummy fields must be created as place-holders.

Record Oriented Data Types	
A – Alphabetic	In – Integer (signed)
AC – Record Oriented Control	IT – Integer (signed)
AN – Alphanumeric	Ln – Zoned decimal with leading sign
Bn – Binary (unsigned)	N – Numeric
BN – Binary unsigned	Nn – Numeric
CH – Character	PD – Packed decimal
DT – Date	Pn – Packed decimal
FN – File name	R – Real (opt. sign and decimal)
Hn – Hexadecimal	TM – Time
HX – Hexadecimal	ZD – Zoned decimal
ID – Identifier	Zn – Zoned decimal

\*n – Number of decimal places implied.

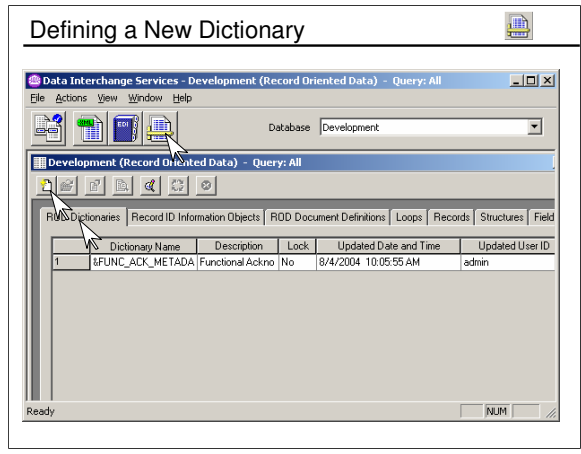
[See table 22 on pages 174-179 of the User's Guide for details.](#)

The translator will automatically convert source to target data types when possible. The most commonly used data types include CH, DT, ID, N, Nn, PD, R, and TM.

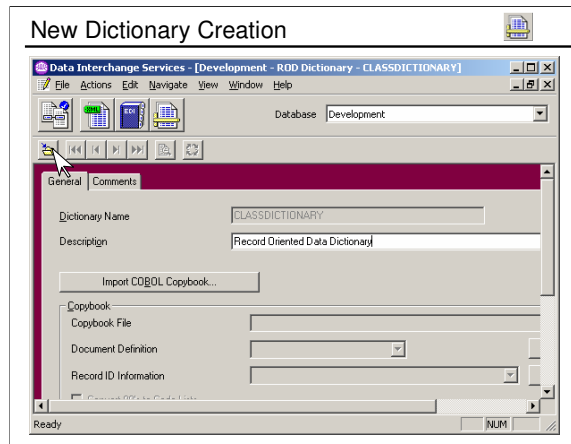
### Creating a New ROD Document Definition

- Obtain the definition of the file layout
- Identify an existing Record Oriented Data Dictionary or create a new one
- Identify an existing Record ID Information object or create a new one
- Define the Record Oriented Data, identifying loops, records, structures, and fields, specifying loop and record repetitions and field types and length

If a Cobol copybook is available, the Record Oriented Document definition may be able to be produced automatically.

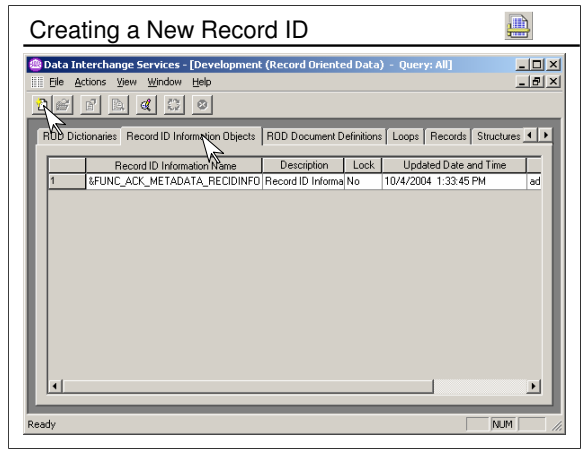


Defining a dictionary does not require defining loops, records, structures, and fields. These are all automatically added to the dictionary as the Record Oriented Document definition is created. Loops, records, structures and fields may be defined manually or modified. Modify these with caution as any change to a field will affect any containing loop, record, or structure as well as any Record Oriented Document definition and associated map using any of these dictionary entries.

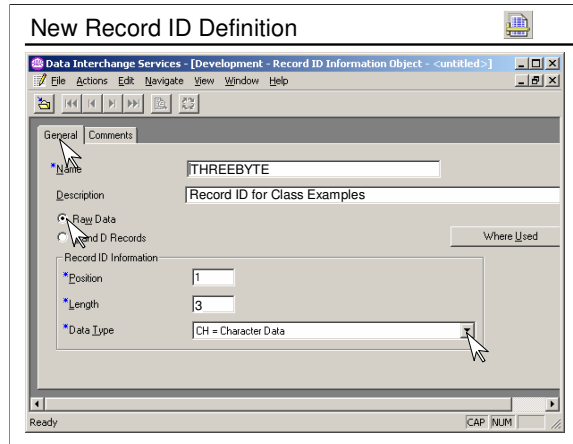


Initially you only need to create an empty Dictionary. Here you assign it a name and optionally a description. Once you save the empty Dictionary, you can identify for use in a Record Oriented Document definition definition. Note that it is at the dictionary tab that a Cobol copybook may imported. The import steps will allow you to create the Record Oriented Document definition during this process. This will be covered later.

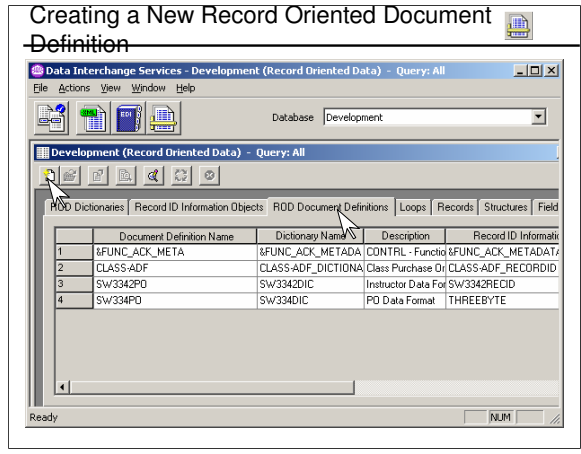




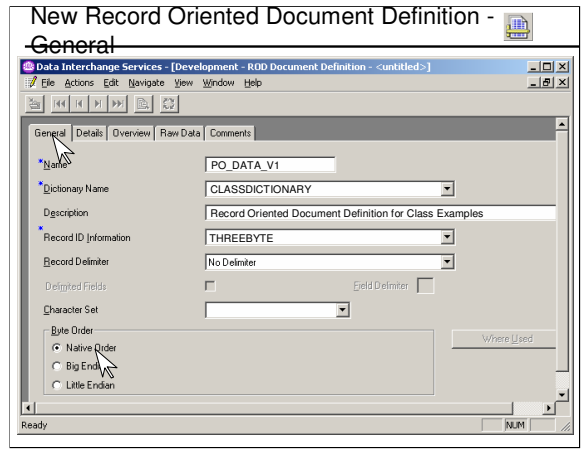
Space for the record ID will need to be reserved in each record, however, the same field name may be used to define each occurrence of the record ID.



The Record ID data type is almost always CH.



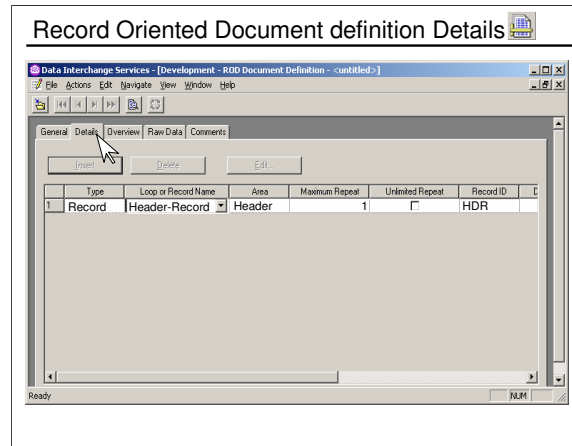
The Record Oriented Document definition tab in the Record Oriented Data functional area holds all Record Oriented Document Definitions for all dictionaries. Before a Record Oriented Document definition can be created, a Dictionary and Record ID Information must be available.



Fields may be fixed-length or comma delimited. In comma-delimited format strings are placed in quotes, numeric values are not. Record Delimiters include: New Line or Carriage Return / Line Feed. Records must be terminated with a record delimiter to support comma delimited fields.

If a code page other than the system default code page is used when reading or writing to a file defined by this Record Oriented Document Definition, select the alternate code page that is to be used.

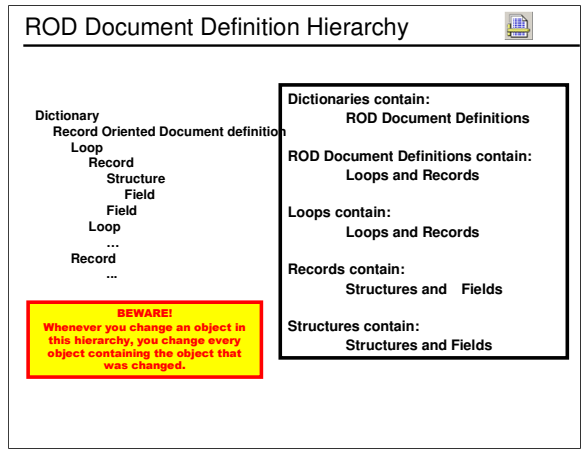
The character set field allows you to select or enter an alternate code page to use if a code page other than the system default code page is



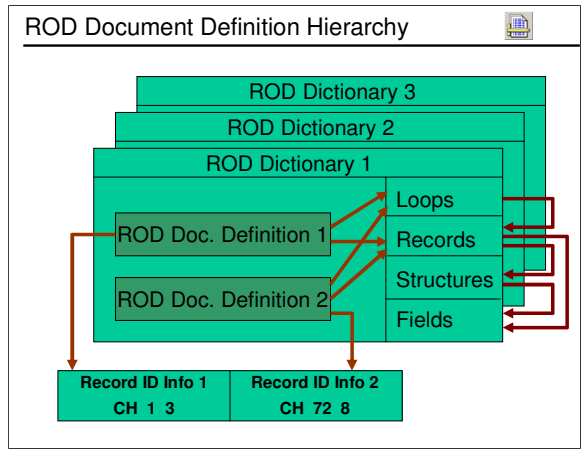
You can create a Record Oriented Document definition by first defining all the highest level objects (Records and Loops) and then open each record or loop to define its contents. Alternatively you can define each record or loop and, after saving it, define its contents before proceeding with the next record or loop definition.

The document area may be specified as “Header”, “Detail”, or “Trailer”. This information is not used during mapping or translation.

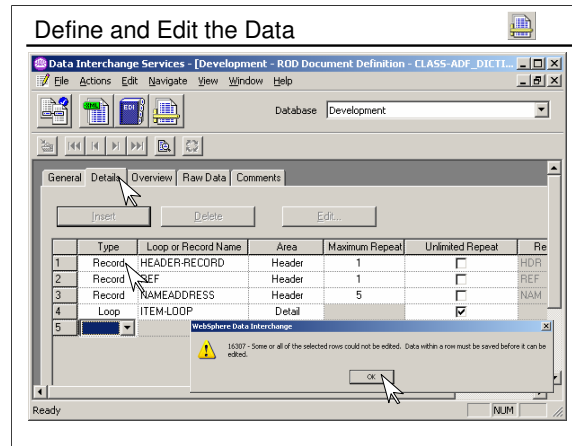
The Record ID is case sensitive and identifies data in the record being defined. This field is identified in the Record ID Information object.



The first level of the document layout can contain Loops and Records. Loops in turn contain Records and other Loops. Records contain Structures and Fields. Structures contain Fields and other Structures. A Field describes a single simple element of data that contains a value. Loops, Records and Structures are considered to be compound elements. A Record defines a complete unit of related information. Records can repeat consecutively. A Loop defines a set of related Records and Loops that repeat as a group.



The first level of the document layout can contain Loops and Records. Loops in turn contain Records and other Loops. Records contain Structures and Fields. Structures contain Fields and other Structures. A Field describes a single simple element of data that contains a value. Loops, Records and Structures are considered to be compound elements. A Record defines a complete unit of related information. Records can repeat consecutively. A Loop defines a set of related Records and Loops that repeat as a group. Loops may contain loops and structures may contain structures.




Here you see three records and one loop defined. No content has yet been defined for these records and loop.

The “Type” column will provide a drop-down list with “Record” and “Loop” available as selections.

The “Loop or Record Name” column will provide a drop-down list with Record and Loop definitions from the dictionary. You can select one from the list or define a new object. If a Record or Loop is selected from the Dictionary, the attributes will be filled in automatically. To change them, the Dictionary entry would need to be redefined.

The “Area” column will provide a drop-down list with “Header”, “Detail”, and “Trailer” available as selections.



Loop/Record Definitions 

**LOOPS/RECORDS**

Type: Record or Loop (Drop Down List)

Loop or Record Name: Identifier (up to 16 characters)

Area: Header/Detail/Trailer – For reference only

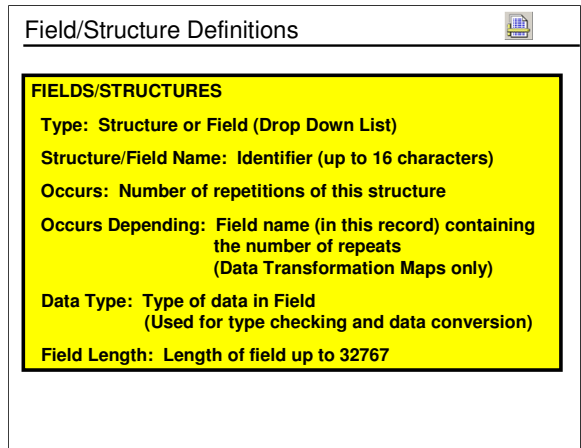
Maximum Repeat: Number of repetitions (or Unlimited)

Unlimited: Check (spacebar) for unlimited repetitions

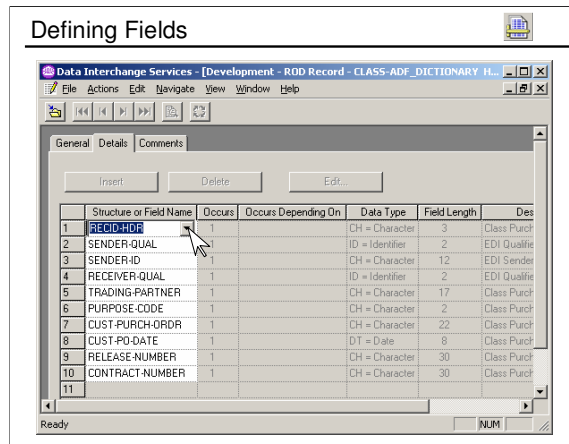
Record ID: Case sensitive character string

Description: For reference only

Attributes of the Record Oriented Data Loop are specified in the ROD Document Definition Editor. The first level of the Loop definition (or layout) is specified. The first level of the Record Oriented Data Loop consists of Records and Loops. Those, in turn, define the remainder of that section of the document. A Record's layout is specified in the ROD Record Editor. Use the ROD Structure Editor to specify the layout of a Structure. The attributes of a Field are specified in the ROD Field Editor.

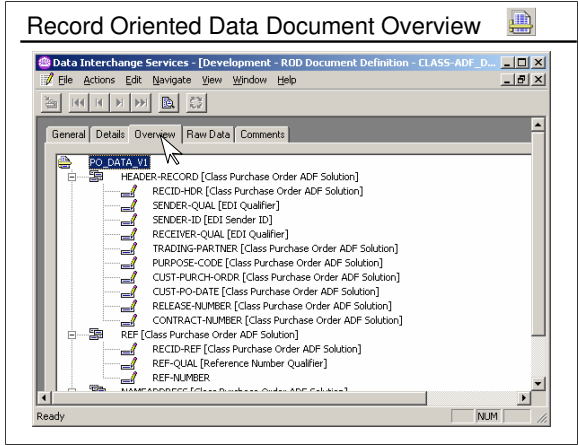


Use the ROD Field Editor to enter new Record Oriented Data Fields into a Record Oriented Data Dictionary or to edit existing Record Oriented Data Fields. From the Field Editor, you can set up WebSphere Data Interchange literals and mapping commands to map data into fields that do not exist in your Record Oriented. Fields are fundamental pieces of data, such as prices or item numbers or first names. In COBOL records, they are stored in a single variable.

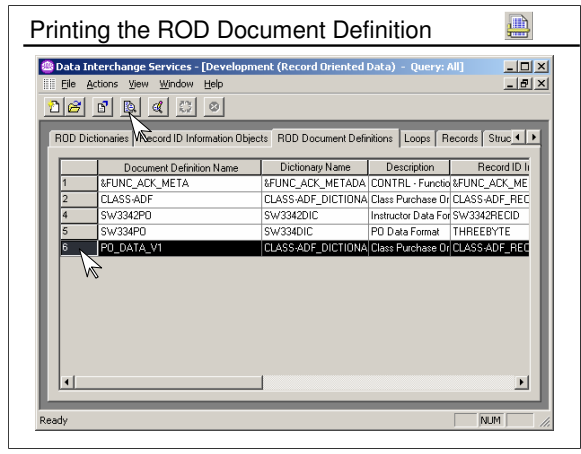


Selections are available from a drop-down list for all columns except comments. The “Structure or Field Name” column will provide a drop-down list with Structure and Field definitions from the dictionary. You can select one from the list or define a new object. If a Structure or Field is selected from the Dictionary, the attributes will be filled in automatically. To change them, the Dictionary entry would need to be redefined.

“Occurs Depending” may be used in Data Transformation maps to indicate the number of times a structure repeats is dependent upon the value of one of other fields in that record.



It is good practice to review the Overview of the Record Oriented Data Document.



You can produce an HTML report of the ROD Document Definition by selecting the document from the list and then selecting the Report icon or selecting *File/Report*.

HTML Report

ROD Document Definition: PO\_DATA\_V1

Document Definition Name	PO_DATA_V1
Dictionary Name	CLASS- ADF_DICTIONARY
Description	Class Purchase Order ADF Solution
Type	Raw Data
Record ID Information Objects	CLASS-            Position 1 Length 3 ADF_RECORDID
Application File or Queue Name	
Application File or Queue Type	
Beginning Record	HEADER-RECORD
Ending Record	

## HTML Report (cont.)

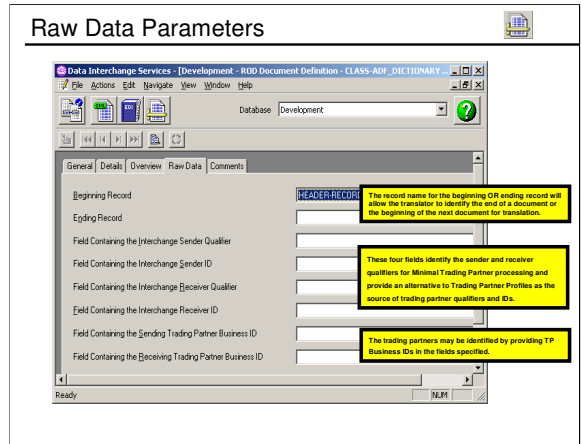
ROD Document Definition: PO\_DATA\_V1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Links Address Google

Record	Record ID	HDR	Maximum Repeat
Record HEADER-RECORD			1
Field	RECID-HDR	Type CH = Character Data	Length 3 Position 1
Field	SENDER-QUAL	Type ID = Identifier	Length 2 Position 4
Field	SENDER-ID	Type CH = Character Data	Length 12 Position 6
Field	RECEIVER-QUAL	Type ID = Identifier	Length 2 Position 18
Field	TRADING-PARTNER	Type CH = Character Data	Length 17 Position 20
Field	PURPOSE-CODE	Type CH = Character Data	Length 2 Position 37
Field	CUST-PURCH-ORDER	Type CH = Character Data	Length 22 Position 39
Field	CUST-PO-DATE	Type DT = Date	Length 8 Position 61
Field	RELEASE-NUMBER	Type CH = Character Data	Length 30 Position 69
Field	CONTRACT-NUMBER	Type CH = Character Data	Length 30 Position 99
Record HEADER-RECORD		Length 128	
Record REF	Record ID REF	Maximum Repeat	1
Field	RECID-REF	Type CH = Character Data	Length 3 Position 1

Done My Computer



The Raw Data parameters are all based on drop-down lists. Note that information on this tab is used only when Raw Data is selected (as opposed to C&D records). The trading partner must be identified as either the sender or receiver. The comments tab can be used provide valuable information. You will find comments tabs throughout Data Interchange Services.



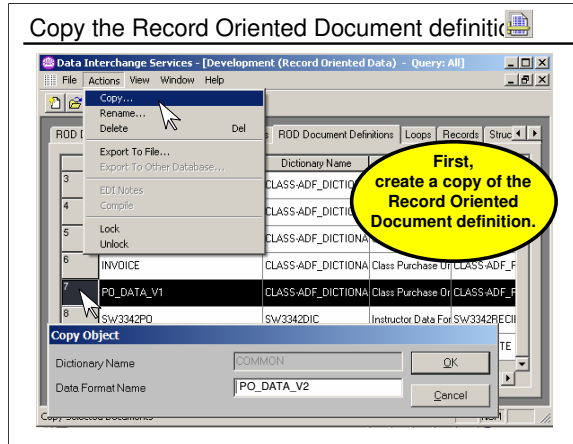
## New Requirements



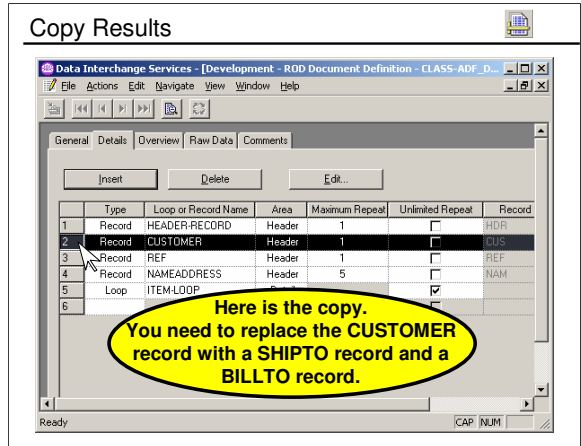
*You have just learned that your trading partners are sending two name loops in the 850 header, one with a bill to (BT) qualifier and one with a ship-to qualifier (ST). You need to create separate records in the Record Oriented Data Document for each of these.*

Once a Record Oriented Data Document Definition is created, it is not unusual to have a requirement to modify that Record Oriented Document definition. The next few slides will review an example and look of some considerations in performing an update to a Record Oriented Document definition.

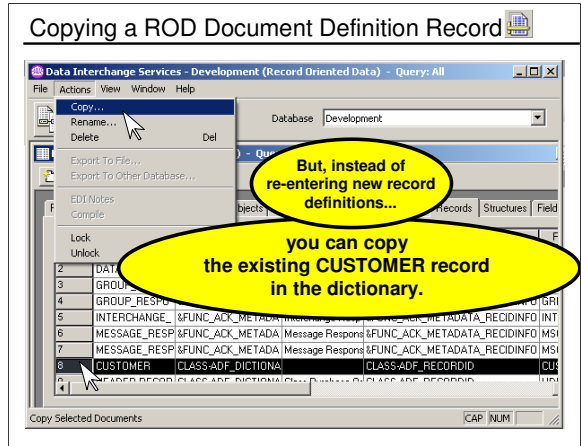
### Copy the Record Oriented Document definition



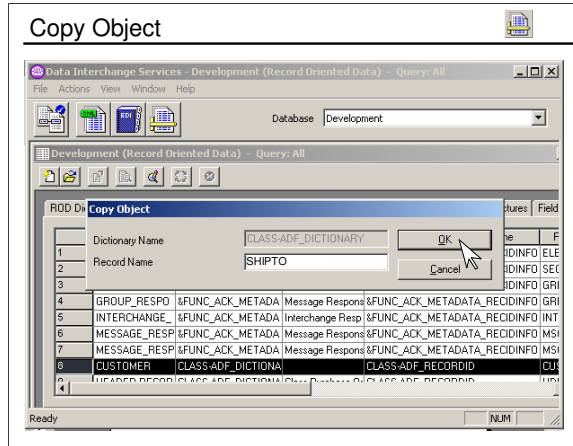
It is a good practice to always create a copy of an object before modifying it.



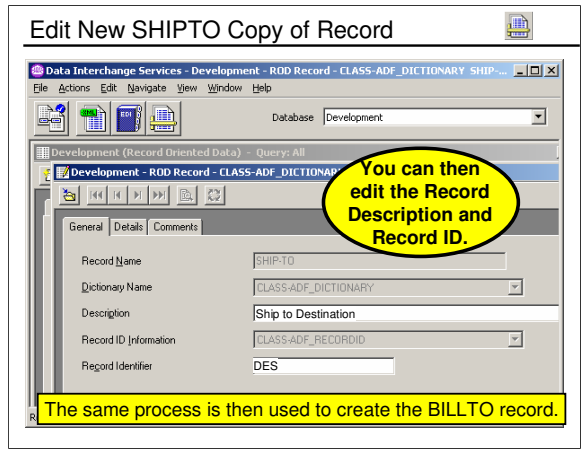
Before deleting the original record, make a copy of it. This eliminates the need to reenter the fields defined for that record.



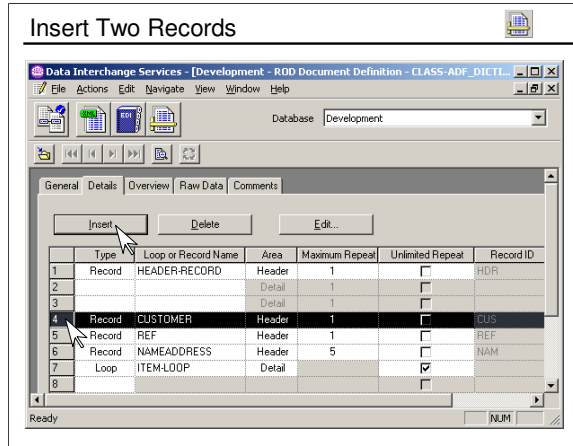
To copy a record, select it (single click), then select Actions/Copy.



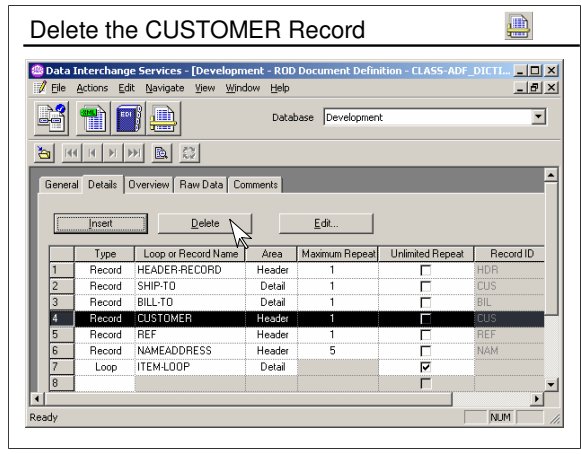
Provide the new name for the copied object.



Be aware of any changes that might need to be made to the definition of the new copies object. Here you need to change the Record ID.

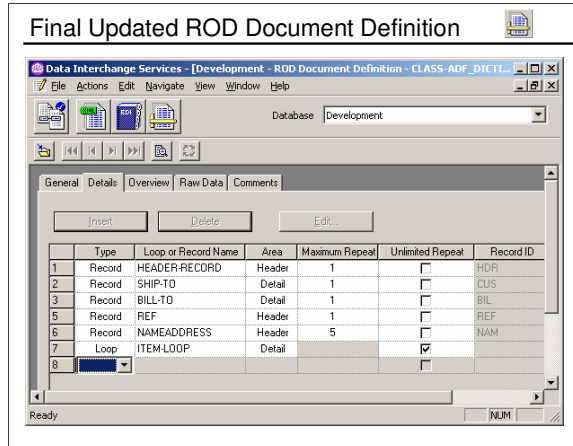


The “Insert” button causes a line to be inserted above the currently selected line. Here you see that two lines have been inserted by clicking “Insert” twice.

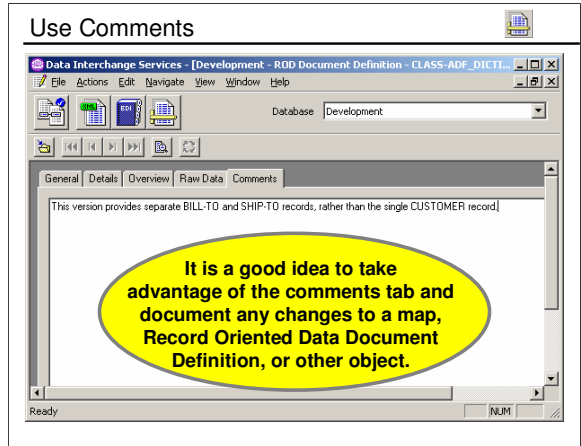


An object may be deleted by clicking the “Delete” button, pressing the Delete key, or selecting Actions/Delete.





This is the modified Record Oriented Document definition with separate SHIP-TO and BILL-TO records replacing the single CUSTOMER record.



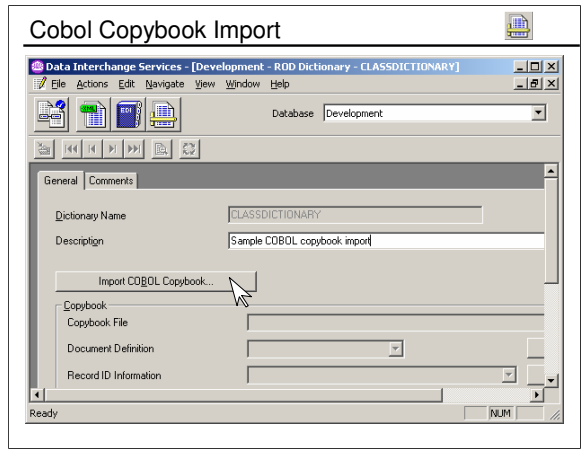
Comments are especially important when changes are made to existing objects.

### Importing a Cobol Copybook

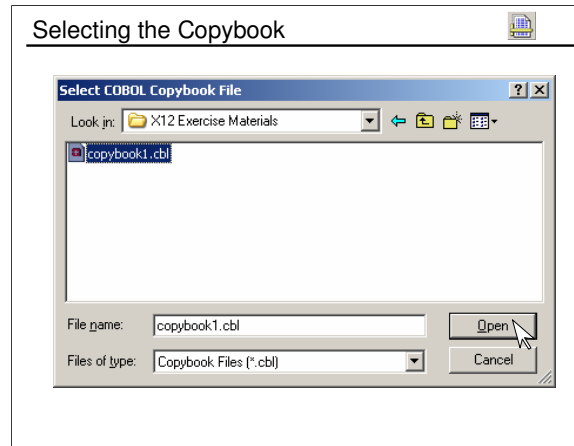


- A Cobol Copybook may be imported into a Record Oriented Data Document definition dictionary to create a Record Oriented Document Definition corresponding to the Copybook
- The Copybook may require some editing prior to importing
- Records must be defined as level 01 in the Copybook
- Following import record IDs will need to be defined for Record Oriented data
- The Record Oriented Document definition produced should be reviewed carefully and manually modified as necessary

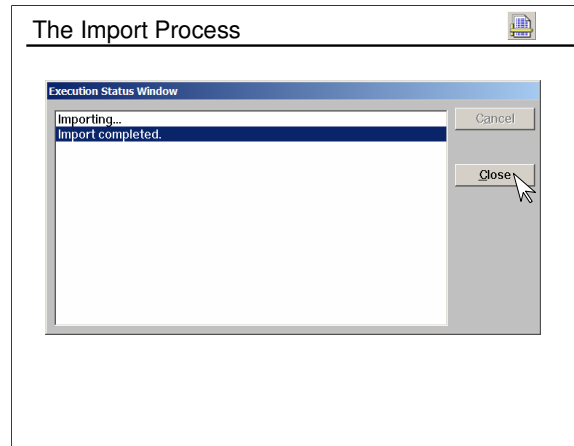
Note that Cobol copybooks are imported under the Dictionary tab, not the Record Oriented Document definition tab.



When a Cobol copybook is imported, initially only the dictionary entries are created. The dictionary must be saved, then a new Record Oriented Document definition created using that dictionary.



If a copybook has been edited, it may have an extension other than “cbl”. This is not a problem for the import process, but you may need to change “Files of type” to browse files with other extensions.



Although it appears that the import is complete, only the dictionary loop, record, structure, and field definitions have been created. You still need to create the Record Oriented Document definition. This is done by selecting “New” for Record Oriented Document Definitions, naming the Record Oriented Document definition, identifying the Record ID Information, and saving the Record Oriented Document definition.

### Unit Summary



- Record Oriented Document definition layouts include Raw Data and C&D records
- Record Oriented Data Dictionaries define all loops, records, structures and fields for one or more Record Oriented Document Definitions
- Record ID Information defines the column, length, and data type of the record identifiers
- Record Oriented Document Definitions are created by defining the loops, records, structures and fields in the data or by importing a Cobol copybook
- Records are defined by identifying the structures and fields in the record
- Structures contain two or more (typically repeating) fields
- Fields are identified as to data type and length

Each of these objectives has been addressed in this unit.