WebSphere® WebSphere Data Interchange for MultiPlatforms

IBM

Version 3.3

Utility Commands and File Formats Reference

SC23-5873-00

WebSphere® WebSphere Data Interchange for MultiPlatforms

IBM

Version 3.3

Utility Commands and File Formats Reference

SC23-5873-00

Note!
Before using this information and the product it supports, read the information in “Notices” on page 325.

March 2007

This edition applies to IBM WebSphere Data Interchange for MultiPlatforms, V3.3. and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this documentation, email doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

About this book

This book is a references of the valid commands, keyword, file formats, and records for WebSphere® Data Interchange.

Who should read this book

This book is intended for anyone that uses commands and file formats in WebSphere Data Interchange.

Syntax conventions used in this book

The following syntax conventions are used throughout this book:

- Bold letters represent values that you must type without change. Unless noted in the text, these values are not case-sensitive. For example:

  EDI PRINT(FILE)

- Bold letters also represent field names from panels. For example:

  Trans data queue

- Lowercase italicized letters represent variable parameters for which you supply the values. For example:

  SYSID(*system-name***)**

Related books

The following books complete the WebSphere Data Interchange library and contain information related to the topics covered in this book. You can view these documents, and download them, from the library page of the WebSphere Data Interchange for MultiPlatforms Web site:

`http://www.ibm.com/websphere/datainterchange`

- *WebSphere Data Interchange for MultiPlatforms Quick Start Guide*, CF0YREN

  This document provides a brief overview of how to use WebSphere Data Interchange.

- *WebSphere Data Interchange for MultiPlatforms Administration and Security Guide*, SC34-6214-01

  This document provides information on administrative tasks you will use in WebSphere Data Interchange.

- *WebSphere Data Interchange for MultiPlatforms Messages and Codes Guide* , SC34-6216-01

  This book provides information to assist you in diagnosing errors.

- *WebSphere Data Interchange for MultiPlatforms User's Guide*, SC34-6215-01

  This book provides information on the WebSphere Data Interchange Client/Server user interface.

- *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01

This document provides detailed technical information about WebSphere Data Interchange.

- *WebSphere Data Interchange for MultiPlatforms Mapping Guide*, SC23-5874-00

  This document provides instructions for your WebSphere Data Interchange mapper.

- *WebSphere Data Interchange for MultiPlatforms Utility Commands and File Formats Reference Guide*, SC23-5873-00

  This document provides the commands and file formats necessary to use WebSphere Data Interchange.

- *WebSphere Data Interchange for z/OS V3.3 Installation Guide*, SC34-6269-01

  This book provides information for the electronic data interchange (EDI) administrator about entering, sending, and receiving EDI transactions and other documents interactively.

Chapter 1. Using The Utility

The WebSphere Data Interchange Utility provides command-level access to WebSphere Data Interchange services. These services can be divided into the following categories:

- Performing general data translation
  - Translating data from any EDI, XML, or data format to any other EDI, XML, or data format using a data transformation map
- Processing outbound documents
  - Translating to EDI standard format using a send map
  - Enveloping
  - Sending
- Processing inbound EDI documents
  - Receiving
  - Deenveloping
  - Translating to data format using a receive map
- Managing data
  - Updating records
  - Removing records
  - Controlling record status
- Reporting and extracting data
  - Formatting and printing reports
  - Printing application data
  - Extracting data
- Customizing
  - Exporting and importing administrative data
- Managing mailboxes (CICS only)
  - Closing mailboxes

Any-to-any data translation

Any-to-any translation is a WebSphere Data Interchange feature that you can use to translate data from any supported source document type to any supported target document type. Supported document types include data formats, EDI standards, and XML data.

This feature expands the flexibility of WebSphere Data Interchange processing by enabling you to process a greater variety of formats. Any-to-any translation uses a *data transformation map*. A data transformation map is a set of mapping instructions that describes how to translate data from a source document into a target document. Both the source and target documents can be any one of several supported document types (application data to EDI standard format, EDI standard format to application data, XML data to EDI standard format, and so on).

Data transformation maps use the TRANSFORM command to do any-to-any translation. You should use any-to-any translation with any maps that you create. This is now the recommended method for translating documents.

Data transformation maps are created using WebSphere Data Interchange client. For more information about the mapping features in WebSphere Data Interchange client, see the *WebSphere Data Interchange for MultiPlatforms User's Guide*, SC34-6215-01 and the .

Outbound processing using send maps

Outbound processing using send maps includes the following functions:

- Translating application data into an EDI standard format and placing it in the Document Store
- Enveloping standard transactions or messages so they are ready to be sent
- Sending enveloped data to trading partners

Commands are supplied to do each of these three steps independently or in combination. In some cases, using the combination commands can improve performance. The following commands are used for outbound processing:

- **ENVELOPE**
- **ENVELOPE AND SEND**
- **RECVFILE AND SEND**
- **REENVELOPE**
- **REENVELOPE AND SEND**
- **RESTART SEND**
- **SEND**
- **SENDFILE**
- **TRANSLATE AND ENVELOPE**
- **TRANSLATE AND SEND**
- **TRANSLATE TO STANDARD**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

With WebSphere Data Interchange you can designate several types of files (print, exception, application, and so on) as MQSeries® Queues.

Fixed-to-fixed translation using send maps

The same outbound processes used for EDI standard translations are also used for fixed-to-fixed translation and enveloping. Fixed-to-fixed translation remains available for use with existing maps in WebSphere Data Interchange for MultiPlatforms V3.3, but all new maps should be translated using any-to-any translation.

When translating to an EDI standard data format, fixed-to-fixed translation is only done when the standard has an envelope type of **F**. The application data being translated can include a mixture of fixed-to-fixed and EDI standard data types. The results of the translation and an image of the data translated are saved to the Document Store in C and D record format and are eligible for enveloping. For more information on the command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

For fixed-to-fixed translation, enveloping and reenveloping transactions from the Document Store includes sorting the transactions to obtain the fewest interchanges and groups, and then writing them to the appropriate file.

The appropriate files to use for EDI standard translated data in order of precedence are the file identified by the `FILEID` keyword, the file specified in the `Trans data queue name` field of the network profile member (with **E** or **U** suffix), or by default, **QDATA** (with **E** or **U** suffix).

The appropriate files to use for EDI standard fixed-to-fixed translated data in order of precedence are the file identified by the `FIXEDFILEID` keyword or the ddname formed by the concatenation of the standard ID (the `Application file name` in the data format that created the standard) with the `File` suffix.

Fixed-to-fixed translated data can either be written to the file in a C and D record format or in raw data format. The D record format is always used when multiple D records are found during translation. This format consists of the `Record ID` with a value of **D**, followed by the 16-byte name of the structure, followed by the structure's data. If the fixed translated data does not have an associated data format, the `Segment ID` from the EDI standard is used as the structure name. For more information on the command syntax and command examples, see Chapter 2, "Utility PERFORM commands," on page 19.

Sending with fixed-to-fixed translation

When the SEND command is performed in a composite command such as ENVELOPE and SEND, WebSphere Data Interchange remembers all the files created during an envelope and sends each file using the appropriate request to the network for sending the data. Standard data is sent using a SEND or SENDFILE command (based on the envelope type used). Fixed translated data with an interchange layer is sent the same way. Fixed translated data without an interchange layer (such as ISA, UNB, and so on) is sent as a file using the SENDFILE command.

You can also use the composite commands such as the TRANSLATE AND ENVELOPE command or ENVELOPE AND SEND command to control the processing of the data you send.

For more information on the command syntax and command examples, see Chapter 2, "Utility PERFORM commands," on page 19.

Inbound processing using receive maps

Inbound processing using receive maps includes the following functions:

- Receiving data from trading partners
- Deenveloping interchanges and placing the EDI standard transactions or messages into the Document Store
- Translating EDI standard transactions or messages into application formats

Commands are supplied to do each of these three steps independently or in combination. In some cases, using the combination commands can improve performance. The following commands are used for inbound processing:

- **DEENVELOPE**
- **DEENVELOPE AND TRANSLATE**
- **RECEIVE**
- **RECEIVE AND DEENVELOPE**
- **RECEIVE AND SEND**
- **RECEIVE AND TRANSLATE**
- **RECVFILE**
- **RESTART RECEIVE**
- **RETRANSLATE TO APPLICATION**
- **TRANSLATE TO APPLICATION**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

In CICS, you use the combination RECEIVE commands when Management Reporting is turned on and SYNCVAL is **-1**. For more information on SYNCVAL, see the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

With WebSphere Data Interchange you can designate several types of files (print, exception, application, and so on) as MQSeries Queues.

TA1 overview

The TA1 is an X12 Interchange Acknowledgment. It acknowledges the receipt and or syntactical correctness of an X12 interchange header and trailer (ISA/IEA) pair. A request to generate a TA1 is based on the value in the ISA14 element. If the ISA14 element (I13) is 1, then an Interchange acknowledgment (TA1) has been requested. If the ISA14 element (I13) is 0, then a TA1 should not be created in response to the ISA. The Interchange control number of a TA1 is matched to a previously transmitted X12 Interchange with the same control number to complete the acknowledgment process. The TA1 segment can appear within an ISA/IEA of its own, or in conjunction with, but prior to, GS/GE segments.

The TA1 segment consists of five separate elements:

Interchange Control Number
Interchange Date (formatted as YYMMDD)\
Interchange Time (formatted as HHMM)
Interchange Acknowledgment Code
A= Accepted
R= Rejected
E= Accepted but the file contains errors and must be resubmitted
Interchange Note Code
000 No error
001 The Interchange Control Number in the header and trailer do not match.

002 This Standard as noted in the Control Standards Identifier is not supported.
003 This Version of the controls is not supported
004 The Segment Terminator is invalid
005 Invalid Interchange ID Qualifier for sender
006 Invalid Interchange Sender ID
007 Invalid Interchange ID Qualifier for receiver
008 Invalid Interchange Receiver ID
009 Unknown Interchange Receiver ID
010 Invalid Authorization Information Qualifier value
011 Invalid Authorization Information value
012 Invalid Security Information Qualifier value
013 Invalid Security Information value
014 Invalid Interchange Date value
015 Invalid Interchange Time value
016 Invalid Interchange Standards Identifier value
017 Invalid Interchange Version ID value
018 Invalid Interchange Control Number
019 Invalid Acknowledgment Requested value
020 Invalid Test Indicator value
021 Invalid Number of Included Group value
022 Invalid control structure
023 Improper (Premature) end-of-file (Transmission)
024 Invalid Interchange Content (e.g., invalid GS Segment)
025 Duplicate Interchange Control Number
026 Invalid Data Element Separator
027 Invalid Component Element Separator
028 Invalid delivery date in Deferred Delivery Request
029 Invalid delivery time in Deferred Delivery Request
030 Invalid delivery time Code in Deferred Delivery Request
031 Invalid grade of Service Code

TA1 processing

To request a TA1 for outbound EDI transactions, the ISA14 element (I13) must be populated with the value 1 by using the **SetProperty** command within the data transformation map, or by using envelope profiles. Pending interchange acknowledgments can be viewed as a report of WebSphere Data Interchange Client. A pending interchange acknowledgment will be identified as a Send Interchange.

When the response TA1 is received for a given interchange, it is reflected or reconciled in the Document Store. TA1 reconciliation is only possible with data transformation processing and if the Document Store is available. If the Document Store is not available, the TA1 segment will be ignored. TA1 reconciliation will not be performed for Interchanges which use control numbers by transaction (document ID), such Interchanges cannot be uniquely identified from information in the TA1 and its encompassing ISA. Only the last TA1 received for a given Interchange will be reflected in the Document Store.

When a TA1 is requested for inbound EDI transactions, the TA1 segment will be generated for incoming EDI transactions regardless of Document Store setting. The TA1 is also generated and enveloped when using delayed functional acknowledgments (when **FADELAY** is set to Y).

WebSphere Data Interchange sets the TA1 acknowledgment code and note code based on ISA/IEA validation accomplished using the SERVICESEGVAL keyword and the validation map $WDI_X44ENV_VAL. The TA1 acknowledgment code of R will be based on the TA1 note code, which is determined by the first error detected during ISA/IEA validation. Only the first error is reported. Interchanges with E errors will be rejected. WebSphere Data Interchange uses the functional acknowledgment map $WDI_TA1_ACK to map the errors reported from the validation map to the TA1 segment.

Considerations

Since the TA1 acknowledges the receipt and syntactical correctness of an X12 Interchange header and trailer (ISA/IEA) pair, the E acknowledgment code cannot be supported. These errors are reported in the functional acknowledgment message.

The following TA1 note codes are not supported during interchange validation:

002 This Standard as noted in the Control Standards Identifier is not supported.
004 The Segment Terminator is invalid
022 Invalid control structure
024 Invalid Interchange Content (for example, invalid GS Segment)
026 Invalid Data Element Separator
027 Invalid Component Element Separator

The TA1 note code 009 is reported with WebSphere Data Interchange Rule not found conditions. Additional validation and reporting are possible by creating and adding mapping commands and code lists based on individual requirements. TA1 note code 020 for Invalid Test Indicator value can result in TA1 reporting code 009 WDI Rule not found condition.

Managing data

Data management includes the following functions:

- Rebuilding interchanges from transactions in the Document Store
- Updating status of transactions in the Document Store
- Removing transactions from the Document Store
- Updating management reporting statistics
- Removing management reporting statistics

The following commands are used for managing data:

- **HOLD**
- **LOAD LOG ENTRIES**
- **PROCESS NETWORK ACKS**
- **PURGE**
- **QUERY**

- **RECONSTRUCT**
- **RECONSTRUCT AND SEND**
- **RELEASE**
- **REMOVE AUDIT TRAIL**
- **REMOVE LOG ENTRIES**
- **REMOVE STATISTICS**
- **REMOVE SUBMISSIONS**
- **REMOVE TRANSACTIONS**
- **RESET STATISTICS**
- **UNLOAD LOG ENTRIES**
- **UNPURGE**
- **UPDATE STATISTICS**
- **UPDATE STATUS**

For more information on command syntax and command examples, see Chapter 2, "Utility PERFORM commands," on page 19.

Removing and archiving event log entries

You can remove event log entries and archive them. Typically, this is a multi-step process because, while event log entries are removed real-time, reclaiming and reorganizing the table space requires an additional step as described in Table 1.

Table 1. Removing and archiving event log entries

Remove DB2® log entries	**PERFORM command**	**Run DBS Utility**
Without archive Without DB2 reorg	REMOVE LOG ENTRIES WHERE APPLID (*value*)	N/A
With archive Without DB2 reorg	UNLOAD LOG ENTRIES WHERE APPLID (*value*) ARCHIVEFILE (*value*) HOLDFILE (*value*)	N/A
Without archive With DB2 reorg	REMOVE LOG ENTRIES WHERE APPLID (*value*)	REORG TABLESPACE EDIELOG
With archive With DB2 reorg	UNLOAD LOG ENTRIES WHERE APPLID (*value*) ARCHIVEFILE (*value*) HOLDFILE (*value*)	REORG TABLESPACE EDIELOG

Reporting and extracting data

Data reporting and extraction includes the following functions:

- Generating formatted reports containing data from the Document Store
- Formatting application data into conventional business documents
- Extracting data from the Document Store for further processing outside WebSphere Data Interchange
- Extracting data from the management reporting statistics tables for further processing outside WebSphere Data Interchange

WebSphere Data Interchange provides two mechanisms for producing reports:
- Management Reporting data extracts
- Document Store data extracts

Both mechanisms collect, update, and extract WebSphere Data Interchange trading partner data and transaction information. Both are invoked with PERFORM commands to extract the data, and both require user-written programs to sort, format, and print data.

The Management Reporting and Document Store Data Extracts differ in the type of information they provide. Management Reporting Data Extracts pull information from the statistics tables, while Document Store Data Extracts pull information from the Document Store.

WebSphere Data Interchange collects information in four management reporting categories and two Document Store reporting categories, as follows:

- Management Reporting Categories
  - Trading Partner Profile

    Provides general trading partner information alphabetically, by trading partner. Reported information includes company name, address, point of contact, telephone numbers, account ID, and user ID.
  - Trading Partner Capability

    Relates trading partner names used. For example, this report determines which maps are installed in test and production, which trading partners are using which translation usages or rules, direction of the translation, EDI standards used, transaction ID, total number of transactions processed, and number of transactions that had errors.
  - Transaction Activity

    Provides information relating transaction volumes to trading partner names. For example, this report can calculate the total number of transactions sent to a trading partner (by date and by map ID) or the total transactions errors by trading partner (and by date or map ID). This information is often used to gauge a trading partner's activity or to reconcile documents sent or received.
  - Network Activity

    Provides information such as network ID, name and account number, user ID, direction, charge code, and total number of bytes sent or received. This information is typically used to determine network charges, by trading partner or application.
- Document Store Reporting Categories
  - Transaction Data Extract

    Extracts detailed technical information from the Document Store. It provides data such as trading partner nickname, direction (S/R), transaction image, send acknowledgment data and image, receive acknowledgment data and image, and transaction handle. This information is commonly used for creating daily reports, such as overdue functional acknowledgments.
  - Envelope Data Extract

Extracts detailed technical information from the Document Store. It provides the same types of information as the transaction data extract except envelope data extract only reports on transactions that have been enveloped, and it sorts the data differently. This information is commonly used for creating daily reports.

– Document Data Extract

Extracts detailed information about transactions, sorted by transaction handle. It provides the ability to extract data from non-EDI documents. This information is commonly used for creating daily reports, such as overdue functional acknowledgments.

PRINT commands

You can use the PRINT commands with selection criteria to extract information from the Document Store and write formatted reports to ddname RPTFILE. To obtain the same type of information without formatting, use the **ENVELOPE** and **TRANSACTION DATA EXTRACT** commands. You can print the information using your system facilities. The cover page for each report lists the selection criteria you entered. You can view most of the information in these reports using the Document Store Facility. Samples of the different formats are listedwith the commands that generate them in “Producing management reports from the Document Store.”

The following commands are used for printing reports:

- **PRINT**
- **PRINT ACKNOWLEDGEMENT IMAGE**
- **PRINT ACTIVITY SUMMARY**
- **PRINT EVENT LOG**
- **PRINT STATUS SUMMARY**
- **PRINT STATUS SUMMARY2**
- **PRINT TRANSACTION DETAILS**
- **PRINT TRANSACTION IMAGE**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

Producing management reports from the Document Store

The following commands are used for extracting data from the Document Store:

- **ENVELOPE DATA EXTRACT**
- **NETWORK ACTIVITY DATA EXTRACT**
- **TRADING PARTNER PROFILE DATA EXTRACT**
- **TRADING PARTNER CAPABILITY DATA EXTRACT**
- **TRANSACTION ACTIVITY DATA EXTRACT**
- **TRANSACTION DATA EXTRACT**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

Creating management reporting reports

To create management reports, do the following:

1. Set the `Management Reporting Active?` field to **Y** in the Application Defaults profile member. The default is **Y**.

   For any trading partner profiles that you want reported, include selection information in the profile fields, such as a department name in a comment field. For more information on management reporting, “TRADING PARTNER PROFILE DATA EXTRACT command” on page 106.
2. Execute the UPDATE STATISTICS command to move the pending information to the statistics tables. You must perform this step prior to data extraction. For more information, “UPDATE STATISTICS command” on page 126.

   ```
   PERFORM UPDATE STATISTICS
   ```
3. Execute the DATA EXTRACT command.

   ```
   PERFORM DATA EXTRACT
   ```

   Together, these two PERFORM commands collect the requested data and place it in a QSAM file named EDIQUERY. You can use wild cards on DATA EXTRACT commands, as follows:

Table 2. Interpretation of wildcard characters

Wild Card	Matches
(any character)	The specified character
?	Any single character
*****	Any sequence of one or more characters

   For more information, see the report examples included with the data extract commands in Chapter 2, “Utility PERFORM commands,” on page 19.
4. Process the data extract file into reports by:
   - Verifying that you successfully created a QSAM data extract file (EDIQUERY) containing a sequential set of independent records with a record length of 1024. For more information, see “Management reporting” on page 307.
   - Sorting the data according to your requirements through a sort utility.
   - Formatting the data using a report-writing utility or a user-written program.
5. Remove previously collected daily statistics by executing the REMOVE STATISTICS command:

   ```
   PERFORM REMOVE STATISTICS
   ```

Because statistics are not dependent on the Document Store, running the REMOVE TRANSACTIONS command does not affect them. For more information, see “PRINT commands” on page 9.

Note: You can use the `USERPGM` parameter to pass the report records to a program prior to writing them to the EDIQUERY QSAM file. When using this parameter, your program should return a code indicating whether the record is written to the EDIQUERY file or is discarded. If a program is not supplied, WebSphere Data Interchange writes the record to the EDIQUERY file.

Creating transaction data or transaction envelope reports

Using the **TRANSACTION DATA EXTRACT** , **ENVELOPE DATA EXTRACT** and **DOCUMENT DATA EXTRACT** commands, you can extract detailed information about transaction data and envelopes from the Document Store, such as:

- Reporting the number of purchase orders sent in a given period of time.
- Reporting the total number of bytes sent in a given period of time. (This can be useful for charging back costs to other departments based on their EDI usage.)
- Creating a customized functional acknowledgment tracking report, selected by application or by department. For example, you can create an Exception Report on Purchase Orders sent more than 2 days ago that have not been acknowledged.
- Creating an exception report flagging missing control numbers for inbound envelopes.
- Creating statistical reports identifying the most frequently used transactions, EDI standards, version sizes, or delimiters.

You can also use the **ENVELOPE DATA EXTRACT** , **TRANSACTION DATA EXTRACT** and **DOCUMENT DATA EXTRACT** commands for functions other than reporting, such as:

- Archiving Document Store data.
- Loading status data directly into the application by application key; for example, the status for each invoice sent can be loaded into the billing system by invoice number.

The **TRANSACTION DATA EXTRACT** command provides data such as trading partner nickname, direction (Send or Receive), transaction image, send acknowledgment data and image, receive acknowledgment data and image, and transaction handle.

The **ENVELOPE DATA EXTRACT** command provides the same types of information as the transaction data extract command except envelope data extracts include only enveloped transactions, sorted by the requested envelope key such as trading partner nickname, direction, interchange control number, receiver ID. Transaction data extracts include all transactions sorted by transaction handle.

The **DOCUMENT DATA EXTRACT** command is an expanded version of the **TRANSACTION DATA EXTRACT** command used to extract data from non-EDI documents. This command extracts detailed information about transactions, sorted by transaction handle.

For more information on report record layouts, see “Transaction data extract record layout” on page 316.

Creating Document Store reports

To create Document Store reports, do the following:

1. Select the type of information to collect in the Document Store:
   a. To collect information, select a Document Store option in the Application Defaults profile other than Do Not Save Information to the Document Store.

b. To collect transaction images, set the Transaction Image field in the Application Defaults profile to a value other than Do Not Save Images of Documents.
c. To collect functional acknowledgment images, set the Functional Acknowledgement Image field in the Application Defaults profile to a value other than Do Not Save Images of Functional Acknowledgements.

Note: For both 1b and 1c, Document Store must be enabled.

2. Perform the usual translations and communications.
3. To extract the information for the reports, execute the appropriate **DATA EXTRACT** command using one of the following commands:
   - PERFORM TRANSACTION DATA EXTRACT
   - PERFORM ENVELOPE DATA EXTRACT
   - PERFORM DOCUMENT DATA EXTRACT
4. Create a user-written program to format the data extract file output (EDIQUERY) into reports.
5. Remove transactions periodically using the following command:

```
PERFORM REMOVE TRANSACTIONS
```

For more information on these commands, see “Reporting and extracting data” on page 7.

Exporting and importing

The **EXPORT** and **IMPORT** commands exports or imports the following:
- Maps (without reworking the entire map)
- Trading partner setup information (such as profile members)
- Administrative data

You tell WebSphere Data Interchange specifically which setup information you want to exchange using a control file.

The following commands are used for exporting and importing:
- **CLOSE MAILBOX**
- **EXPORT**
- **IMPORT**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

Profile maintenance

You can query for specific profiles to review them and you can delete obsolete profiles. The following commands are used for exporting and importing:
- **DELETE PROFILE**
- **QUERY PROFILE**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

Continuous receive

WebSphere Data Interchange can take advantage of the continuous receive capabilities provided by GXS Expedite/CICS. With continuous receive, you will receive data immediately after it is put in the VAN mailbox. The continuous receive function is only available on CICS.

When a continuous receive is started through WebSphere Data Interchange, WebSphere Data Interchange and GXS Expedite/CICS each keep a control record about the continuous receive. Although WebSphere Data Interchange and GXS Expedite/CICS can get out of sync when managing continuous receives, this is an exception to normal processing. For more information about out of sync situations, the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

The following commands are used for continuous receive:

- **REPORT CONTINUOUS RECEIVE STATUS**
- **START CONTINUOUS RECEIVE**
- **STOP CONTINUOUS RECEIVE**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

Reporting continuous receive status

A continuous receive status report includes the status of either a single continuous receive member or all continuous receives known to WebSphere Data Interchange and GXS Expedite/CICS. The output goes to the report file specified for the report file name and type in the Utility Control Information Block. The default is RPTFILE (type **TS** in CICS). If an error is encountered while generating the report, the processing program immediately terminates and does not report on subsequent continuous receives.

Note: Continuous receive report statuses can be good or bad depending on the result you expected. For example, a status of EXP STARTED N/P can be an acceptable response that perfectly describes the state of that particular continuous receive. On the other hand, a status of EXP STARTED might be a problem and indicate an out-of-sync situation between WebSphere Data Interchange and GXS Expedite/CICS.

Table 3. Continuous receive status codes

If report status is:	This continuous receive is:
STARTED	Considered to be running by both WebSphere Data Interchange and GXS Expedite/CICS. A continuous receive profile member exists.
DI STARTED	Considered to be running by WebSphere Data Interchange, but not by GXS Expedite/CICS. A continuous receive profile member exists.

Table 3. Continuous receive status codes (continued)

If report status is:	This continuous receive is:
EXP STARTED	Considered to be running by GXS Expedite/CICS, but not by WebSphere Data Interchange. A continuous receive profile member exists.
STARTED N/P	Considered to be running by both WebSphere Data Interchange and GXS Expedite/CICS. However, a continuous receive profile member does not exist.
DI STARTED N/P	Considered to be running by WebSphere Data Interchange, but not by GXS Expedite/CICS. A continuous receive profile member does not exist.
EXP STARTED N/P	Considered to be running by GXS Expedite/CICS, but not by WebSphere Data Interchange. A continuous receive profile member does not exist.
NOT STARTED	Not considered to be running by either WebSphere Data Interchange or GXS Expedite/CICS. However, a continuous receive profile member does exist and it is eligible to be started.
NOT STARTED N/A	Not considered to be running by either WebSphere Data Interchange or GXS Expedite/CICS. A continuous receive profile member does exist, but it is not eligible to be started.

Table 4. Continuous receive status report record format

Name	Offset	Length	Type	Description
CRSTATUS	0	16	Char	Report status
CDPROFID	16	16	Char	WebSphere Data Interchange continuous receive profile member
CDREQACT	32	08	Char	WebSphere Data Interchange requestor's account
CDREQUSE	40	08	Char	WebSphere Data Interchange requestor's user ID
CDUNIQUE	48	08	Char	WebSphere Data Interchange continuous receive unique ID
CDMSGUC	56	08	Char	WebSphere Data Interchange message user class
CDTPACCT	64	08	Char	WebSphere Data Interchange trading partner's account
CDTPUSER	72	08	Char	WebSphere Data Interchange trading partner's user ID
CEREQACT	80	08	Char	GXS Expedite/CICS account
CEREQUSE	88	08	Char	GXS Expedite/CICS user ID
CEUNIQUE	96	08	Char	GXS Expedite/CICS continuous receive unique ID
CEMSGUCL	104	08	Char	GXS Expedite/CICS message user class
CETPACCT	112	08	Char	GXS Expedite/CICS trading partner's account

Table 4. Continuous receive status report record format (continued)

Name	Offset	Length	Type	Description
CETPUSER	120	08	Char	Expo/CICS trading partner's user ID
RESERVED	128	04	Char	Blanks

Persistent environment

The persistent environment is an optional WebSphere Data Interchange feature available with CICS/ESA. This command is available only in the CICS environment and is not required for normal processing. This command is used to gather information required by WebSphere Data Interchange support personnel when debugging problems related to the Persistent Environment. The resulting dump is written to EDIGDMP1 in the CICS startup JCL.

The following commands are used for debugging the persistent environment:
- **GLB DUMP**
- **GLB TRACE**

For more information on command syntax and command examples, see Chapter 2, “Utility PERFORM commands,” on page 19.

Using the WebSphere Data Interchange Utility in the z/OS environment

WebSphere Data Interchange provides sample JCL statements for running the DataInterchange Utility in z/OS. You can copy and customize these statements as necessary. The following is an explanation of the parameters your application can pass in the JCL. Sample JCL files are shipped with the DataInterchange Host product. All the parameters are DataInterchange keywords and optional (except PLAN and SYSTEM, which can be required under certain conditions).

APPLID=*aaaaaaaa*

Specifies the application ID to run the DataInterchange Utility. This parameter also identifies the log file specified in the Application Default (APPDEFS) profile. Replace *aaaaaaaa* with the ID of the application that initialized DataInterchange. If you specify this parameter, the Application Default profile must contain a matching entry to define which log file is used for recording errors and events pertaining to the application. The two `APPLID` values shipped with DataInterchange are:
- **EDIFFS (default)**. Associated with the LOGFFS ddname. The default APPLID and log when using the utilities
- **EDIMP**. Associated with the LOGEDI ddname. The APPLID and log used during online DataInterchange processing

SYSID=*bbbbbbbb*

Identifies the installation-defined WebSphere Data Interchange system used to run the EDIUTIL utility. Specifying this parameter controls access to various components of DataInterchange. The `SYSID` is part of the resource name

defined using RACF or some other resource control product. Replace *bbbbbbbb* with the ID used to protect DataInterchange services (for example, RACF). The default ID is **DIENU**.

LANGID=*ccc*

Identifies the language ID used to run the EDIUTIL utility. The value supplied must match an entry in the language(LANGPROF) profile. The language ID is used to establish values such as date formats and decimal notation. Replace *ccc* with the following value for the language version:

ENU. English

DLM=*d*

Specifies the delimiter used in place of left and right parentheses to enclose values in the EDIUTIL utility command language. Replace *d* with the delimiter you want to use in place of the left and right parentheses to enclose values in the DataInterchange Utility command language. You must supply this parameter if a keyword value contains either a left or right parenthesis.

MQSYSIN=*eeeeeeee*

Replace *eeeeeeee* with the DataInterchange MQSeries Queue profile member you want to use instead of a sequential file allocated to either ddname SYSNAME or ddname EDISYSIN.

MQPRT=*ffffffff*

Replace *ffffffff* with the DataInterchange MQSeries Queue profile member you want to use instead of a sequential file allocated to ddname PRTFILE.

MQRPT=*gggggggg*

Replace *gggggggg* with the DataInterchange MQSeries Queue profile member you want to use instead of a sequential file allocated to ddname RPTFILE.

MQEXCP=*hhhhhhhh*

Replace *hhhhhhhh* with the DataInterchange MQSeries Queue profile member you want to use instead of a sequential file allocated to ddname FFSEXCP.

MQTRAK=*iiiiiiii*

Replace *iiiiiiii* with the DataInterchange MQSeries Queue profile member you want to use instead of a sequential file allocated to ddname FFSTRAK.

MQQUERY=*jjjjjjjj*

Replace *jjjjjjjj* with the DataInterchange MQSeries Queue profile member you want to use instead of a sequential file allocated to ddname EDIQUERY.

PLAN=*kkkkkkkk*

In a DB2 environment, if the utility is invoked by way of IKJEFT01 and there is no EDITSIN data set allocated, then this parameter is required. For more information, see *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01. Replace *kkkkkkkk* with the DB2 plan name.

SYSTEM=*llll*

In a DB2 environment, if the utility is invoked by way of IKJEFT01 and there is no EDITSIN data set allocated, then this parameter is required. For more information, see *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01. Replace *llll* with the name of the DB2 subsystem (or group, if Data Sharing).

Optional JCL parameter example

```
PARM('SYSID=TEST LANGID=ENU SYSTEM=DB93 PLAN=EDIENU32')
```

Chapter 2. Utility PERFORM commands

This chapter explains how to use the WebSphere Data Interchange Utility command language. For additional information about invoking the WebSphere Data Interchange Utility in an z/OS or AIX environment, see the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

Command language syntax

The WebSphere Data Interchange Utility command language consists of PERFORM statements, WHERE clauses, and SELECTING clauses. It is represented in the text as follows:

```
PERFORM SAMPLE-COMMAND
    SELECTING SAMPLE-KEYWORD(value)
    WHERE SAMPLE-KEYWORD(value)
```

A PERFORM statement defines the action WebSphere Data Interchange takes. The following statement consists of the word PERFORM followed by the TRANSLATE TO APPLICATION command as follows:

```
PERFORM TRANSLATE TO APPLICATION
```

While your input can include more than one PERFORM statement, the WebSphere Data Interchange Utility handles each statement separately. It verifies the syntax of each statement and processes it before verifying and processing the next statement. If a nonzero (error) return code is generated for a statement, processing stops with the incorrect statement.

A WHERE clause supplies selection criteria and other information that WebSphere Data Interchange requires to process your requests. A WHERE clause consists of the word WHERE, followed by one or more keywords and their associated values. You must enclose the associated values in parentheses. For example, see the keyword value **PISCES** in the WHERE clause:

```
    WHERE TPNICKN(PISCES)
```

Note: In z/OS, you can redefine the parentheses delimiter with the `DLM` parameter. For more information, see “Using the WebSphere Data Interchange Utility in the z/OS environment” on page 15.

A PERFORM statement can include more than one WHERE clause. Each WHERE clause can contain more than one keyword, but each keyword can be used only once within a WHERE clause. You can use a keyword again on the same PERFORM statement in a different WHERE clause. The following example shows a command statement that uses a WHERE clause:

```
PERFORM TRANSLATE TO APPLICATION
    WHERE TPNICKN(PISCES) TRXDATE(12/12/01)
    WHERE TPNICKN(PISCES) TRXDATE(12/14/01)
```

The previous statement tells WebSphere Data Interchange to translate to application format all EDI documents that were received from a trading partner whose name is PISCES, and whose documents were placed in the Document Store on December 14, 2001. WebSphere Data Interchange processes only those documents that meet both of the conditions specified in the WHERE clause. When you specify all of your conditions in one WHERE clause, WebSphere Data Interchange limits processing to only those transactions that satisfy all the specified conditions.

If you want WebSphere Data Interchange to process EDI documents with either condition, use two separate WHERE clauses. For example, the following statement will translate all records for trading partner PISCES as well as all records with a transaction date of December 14, 2001:

```
PERFORM TRANSLATE TO APPLICATION
    WHERE TPNICKN(PISCES)
    WHERE TRXDATE(12/10/01)
```

Some statements use a SELECTING clause. Like WHERE clauses, SELECTING clauses provide selection criteria. However, you can only include one SELECTING clause in each PERFORM statement. The following example shows a command statement that uses a SELECTING clause:

```
PERFORM ENVELOPE DATA EXTRACT
    SELECTING INTERCHANGE(Y) GROUP(Y) TRANSACTION(Y)
    WHERE TPNICKN(PISCES) INTCTLNO(000008888) DIR(S)
```

The command language format is free-form. You can insert blanks between keywords to improve readability, but do not insert blanks between a keyword and its associated value. Characters can be in upper, lower, or mixed case. Most fields are not case sensitive. To determine if a field is case sensitive, see the field's definition in Chapter 3, "Utility keywords," on page 129.

To select a range of values, you can follow some keywords with a pair of values separated by the keyword `TO`. The first value indicates the low end of the range. The second value indicates the high end of the range. Only transactions with values falling between these two values will be selected.

To insert comments between commands, place an asterisk (*) in column one and type the text on the same line. Each line of a comment must begin with an asterisk or the system will attempt to process the information as commands. For example:

```
//SYSIN DD *
*
*This command translates my data into standard format
*
PERFORM TRANSLATE TO STANDARD
    WHERE APPFILE(APDATA01)
/*
```

DATE, TIME, and HANDLE keywords

Several keywords are used to identify date or time. The format of the date and time values you submit must conform to the Date mask and Time mask fields in the

language profile used. You can request the current date by using an asterisk (*), or you can request a previous date by using *-*n*, where *n* is the number of days before today's date. You can also use an asterisk to request the current time.

The HANDLE keyword is used to select a specific transaction or transaction group for processing or reporting. This keyword specifies the date, time, and transaction ID in format YYYYMMDDHHMMSSnnnnnn.

Command language validation

WebSphere Data Interchange performs basic validation on the command language input, as follows:

- Each PERFORM command (such as TRANSLATE TO STANDARD) is validated. WebSphere Data Interchange generates an error if the command is not valid.
- All keywords are validated. WebSphere Data Interchange generates an error if a keyword is not valid.
- If a keyword is specified, an associated value must also be specified. WebSphere Data Interchange generates an error if you do not specify a keyword value.
- You must enclose each keyword value with beginning and ending delimiters. The default delimiters are opening and closing parentheses. WebSphere Data Interchange generates an error if you do not enclose a keyword value with delimiters.
- The length of each keyword value is validated against the maximum length for that keyword. WebSphere Data Interchange generates an error if the length of the value you supply is greater than the maximum length for the keyword.
- Keywords cannot be used more than once in a WHERE clause. However, you can specify multiple WHERE clauses. WebSphere Data Interchange generates an error if you specify a keyword more than once in the same WHERE clause.
- Date and time fields are validated. WebSphere Data Interchange generates an error if the date or time values you specify are not valid.
- When a keyword has an associated default value, keyword values that are not valid are ignored, and defaults are used. An example of a keyword value that is not valid is `RAWTEST`(**X**). In this situation, WebSphere Data Interchange ignores the value of **X** and does not generate an error. For specific default values, see Chapter 3, “Utility keywords,” on page 129.
- If a keyword is used as part of selection criteria against the Document Store, the associated value is accepted. If the associated value is not valid, WebSphere Data Interchange uses the value as part of the selection criteria, and does not find a transaction match. Verify the associated values for each keyword used for Document Store selection criteria.

Error filtering

You can use a special named variable called DIERRFILTER to tell the translator to ignore certain error conditions. DIERRFILTER can also be used as a keyword on PERFORM commands to indicate which errors should be ignored during the translation process.

When the DIERRFILTER variable is specified as **SET** or **SAVED**, the translator parses the value of the variable as an indicator of the errors to ignore. The DIERRFILTER variable value should consist of a list of the error codes to be ignored. These values are documented in the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

When an error is filtered, it does not mean the error did not occur. Eliminating the message and its corresponding condition code does not eliminate the error. For example, a TR0004 error message can be issued if a validation edit fails. This error can be filtered so as not to produce the TR0004 message, but doing so does not alter the fact that the edit failed. Filtering only serves to suppress the error message and its condition code but does not otherwise affect translation processing.

Note: If you filter an error that would normally generate a functional acknowledgment, filtering the error eliminates the functional acknowledgment as well. If the error is not reported because of filtering, it is not reported in a functional acknowledgment either.

For example, if you are not interested in the “mandatory data element missing” error (TR0001-101), or the “data element is too short” error (TR0003-103), you can filter them out by specifying:

```
&SET DIERRFILTER 101,103
```

You can filter many errors without typing a code for each one by specifying a range of values in the variable. For example, you would eliminate errors 101, 103 and all warning messages by specifying:

```
&SET DIERRFILTER 101,103,1-99
```

Because DIERRFILTER is a named variable, you can use a statement like the following to eliminate validation errors, along with those errors previously set:

```
&SET DIERRFILTER &E(DIERRFILTER + ',116')
```

Certain errors occur before translation is performed, even before a map has been determined. Most of these errors fall into the warning category. You can use the following PERFORM command to eliminate all the warning errors before translation is performed:

```
PERFORM TRANSLATE TO STANDARD
    WHERE DIERRFILTER(1-99)
```

The DIERRFILTER variable has a transaction scope and, therefore, the value set on one transaction does not propagate to the next transaction. The translator sets the DIERRFILTER variable to the value specified in the PERFORM command at the start and end of each transaction.

There are certain errors that you cannot turn off. For example, you cannot filter error TR0053 (maximum number of structures exceeded) because the message is too important, as it indicates data is being ignored. If you attempt to filter an error that is not filterable, the request is ignored (no error message is generated).

Errors TR0101, TR0103, TR0151, TR0153, TR0203 and TR0205 indicate that the ISA/IEA, GS/GE, or ST/SE segment pairs are inconsistent with respect to control numbers or control counts. If these errors are filtered, WebSphere Data Interchange will process the interchange, group, or message in spite of the inconsistency.

In the case where a map contain a complex set of DIERRFILTER values, and something is not working correctly, you might need to see all error messages. Rather than changing the map, you can use the DIERRFILTER keyword with a value of **IGNORE** to tell WebSphere Data Interchange that all errors should be reported regardless of the instructions in the map, as follows:

```
PERFORM TRANSLATE TO STANDARD
    WHERE DIERRFILTER(IGNORE)
```

Overriding utility condition codes

The WebSphere Data Interchange Utility returns a condition code to z/OS that can be tested in the JCL. In cases where you want to ignore certain condition codes, you can use the keywords IFCC and SETCC on any PERFORM statement to override up to 10 different condition codes.

Note: When overriding condition codes, do not ignore important error conditions.

These keywords are entered as follows:

- IFCC(*c1*,*c2*,*c3*,*c4*,*c5*,*c6*,*c7*,*c8*,*c9*,*c10*)

  The values *c1-c10* represent the utility condition codes to be checked.
- SETCC(*n1*,*n2*,*n3*,*n4*,*n5*,*n6*,*n7*,*n8*,*n9*,*n10*)

  The values *n1-n10* will be used to override the IFCC condition codes *c1-c10*, respectively.

If SETCC keyword is not specified, all codes specified in the IFCC keyword are overridden to zero (`0`).

The following is an example of an override of an empty application file condition on TRANSLATE TO STANDARD:

```
PERFORM TRANSLATE TO STANDARD
    WHERE APPFILE(APDATA01) IFCC(6) SETCC(0)
```

Note: Since zero (0) is the default override code, you can omit the `SETCC` keyword and get the same results as entering a value of zero.

CLOSE MAILBOX command

WebSphere Data Interchange does not wait for send requests to complete when using GXS Expedite/CICS and Information Exchange. When any type of access to Information Exchange is accomplished (send, receive, continuous receive, network status update), the mailbox opens and remains open until you close it. Closing your mailbox is only necessary when the same mailbox will be used in the z/OS environment. To close the mailbox, you must use this command to end the session. This command is only available in the CICS environment and is not required for general processing.

Syntax

CLOSE MAILBOX
REQID(*requestor ID*)

Example

Close the mailbox for requestor **DEPTA7F**.

```
PERFORM CLOSE MAILBOX
    WHERE REQID(DEPTA7F)
```

DEENVELOPE command

This command takes EDI transactions from the envelope file, removes the envelope segments, and places the results in the Document Store. The envelope file is either the receive file specified in the mailbox (requestor) profile or the file specified in the command.

Syntax

DEENVELOPE
DIERRFILTER(*initial error filter set*)
DUPENV(*process duplicate envelopes*)
EXTENDC(*translate with extended C record format*)
FADELAY(*delay queuing functional acknowledgment*)
FILEID(*processing file ddname*)
FORCETEST(*force test usage*)
FUNACKFILE(*functional acknowledgment ddname*)
FUNACKREQ(*require functional acknowledgment envelope file*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
MRREQID(*management reporting requestor ID*)
MULTIDOCS(*multiple-document file*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RECOVERY(*recovery unit of work*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
XML(*XML required*)
XMLDICT(*XML dictionary address*)
XMLDTDS(*XML DTD path*)
XMLEBCDIC(*EBDCDIC indicator)*
XMLSEGINP(*line break indicator*)
XMLSTDID(*destination EDI standard ID*)
XMLVALIDATE(*XML validation indicator)*

Example 1

Deenvelope the transactions previously received for requestor ID **DEPTA7F**. Place the results in the Document Store. Mark these transactions for purging after 10 days in the store.

```
PERFORM DEENVELOPE
    WHERE REQID(DEPTA7F) PURGINT(10)
```

Example 2

Deenvelope the transactions in ddname **A7FIN**, place the results in the Document Store and do not process duplicate interchanges.

```
PERFORM DEENVELOPE
    WHERE FILEID(A7FIN) DUPENV(N)
```

DEENVELOPE AND TRANSLATE command

This command combines the functions of the DEENVELOPE and TRANSLATE TO APPLICATION commands. It takes EDI documents from the receive file defined in the mailbox (requestor) profile member or from the override file specified in the command, removes the envelope segments, and places the results in the Document Store. It then translates the documents to the defined application format and places the results in the application file specified by the data format or in the file specified by the trading partner usage/rule. This composite command performs faster than the separate commands.

Note: In CICS, you can also deliver the data to a program of CICS transaction, as specified in the `Application file name` and `Application file type` fields of the data format.

Syntax

DEENVELOPE AND TRANSLATE
ASSERTLVL(*session assertion level*)
BATCHSET(*set transaction batch ID*)
CCEXCEPTION(*job-step condition code*)
DIERRFILTER(*initial error filter set*)
DUPENV(*process duplicate envelopes*)
EXTENDC(*translate with extended C record format*)
FADELAY(*delay queuing functional acknowledgment*)
FILEID(*processing file ddname*)
FORCETEST(*force test usage*)
FUNACKFILE(*functional acknowledgment ddname*)
FUNACKREQ(*require functional acknowledgment envelope file*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
MRREQID(*management reporting requestor ID*)
MULTIDOCS(*multiple-document file*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RAWDATA(*translate to raw data format*)
RECOVERY(*recovery unit of work*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
XML(*XML required*)
XMLDICT(*XML dictionary address*)
XMLDTDS(*XML DTD path*)
XMLEBCDIC(*EBDCDIC indicator)*

XMLSEGINP(*line break indicator*)
XMLSTDID(*destination EDI standard ID*)
XMLVALIDATE(*XML validation indicator)*

Example 1

Deenvelope and translate the EDI documents in the receive file specified by requestor ID **DEPTA7F**.

```
PERFORM DEENVELOPE AND TRANSLATE
    WHERE REQID(DEPTA7F)
```

Example 2

Deenvelope and translate the EDI documents in the receive file specified by requestor ID **DEPTA7F** into raw data format. Return the information records to the exception file.

```
PERFORM DEENVELOPE AND TRANSLATE
    WHERE REQID(DEPTA7F) RAWDATA(Y) OPTRECS(I)
```

DELETE PROFILE command

This command deletes profile members. However, you delete a trading partner profile member while it is still associated with a translation usage/rule.

Syntax

DELETE PROFILE
ID(*DataInterchange profile name*)
MEMBER(*member profile name*)

Example 1

Delete trading partner profile member **TPMEM**.

```
PERFORM DELETE PROFILE
    WHERE ID(TPPROF) MEMBER(TPMEM)
```

Example 2

Delete mailbox (requestor) profile member **REQMEM**.

```
PERFORM DELETE PROFILE
    WHERE ID(REQPROF) MEMBER(REQMEM)
```

DELETE USAGE command

This command deletes send or receive usages. Default for DIR is 'R'. If both APPRECID and APPSNDID are specified, APPSNDID will be used. Any management reporting statistics related to the Usages deleted will also be deleted.

Syntax

DELETE USAGE
AGENCY(*Responsible agency code, 8 characters, default is blank*)
APPLTPID(*Application Trading Partner nickname*)
APPRECID(*Application Receiver ID, 35 characters*)
APPSNDID(*Application Sender ID, 35 characters*)
DIR(*S(send) or R(receive), default is R*)
REL(*Release, 8 characters, default is blank*)
TPID(*Internal Trading partner ID, 35 bytes*)
TPNICKN(*EDI Trading Partner Nickname*)
MAPID(*Mapname, 30 characters*)
VERSION(*Version, 8 characters, default is blank*)

Example 1

Delete send usage on map **mapname** where application trading partner nickname is **atpid** and edi trading partner nickname is **etpid** and internal trading partner id is **intpid**.

```
PERFORM DELETE USAGE
WHERE DIR(S) MAPID(mapname) APPLTPID(atpid)  TPNICKN(etpid) TPID(intpid)
```

Example 2

Delete receive usage on map **mapname** where application trading partner nickname is **atpid** and edi trading partner nickname is **etpid** and application sender id is **appsenderid** and agency is **agency** and version is **version** and release is **release**.

```
PERFORM DELETE USAGE WHERE DIR(R) MAPID(mapname) APPLTPID(atpid)
TPNICKN(etpid) APPSNDID(appsenderid)
AGENCY(agency) VERSION(version) REL(release)
```

Example 3

Delete all usages for a given map.

```
PERFORM DELETE USAGE WHERE DIR(S/R) MAPID(mapname)
```

DOCUMENT DATA EXTRACT command

This command extracts detailed information about non-EDI documents, sorted by transaction handle. You can use this command to create report data to:

- Report on the number of purchase orders sent in a given period of time
- Report on the total number of bytes sent in a given period of time (this can be useful for charging back costs to other departments based on their EDI usage)

You can also use this command for functions other than reporting, such as:

- Archiving Document Store data
- Loading status data directly into the application by application key; for example, the status for each invoice sent can be loaded into the billing system by invoice number

Syntax

DOCUMENT DATA EXTRACT
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
ACFIELD1(*starting application control field data*)
TO(*ending application control field data*)
ACFIELD2(*starting application control field data*)
TO(*ending application control field data*)
ACFIELD3(*starting application control field data*)
TO(*ending application control field data*)
ACFIELD4(*starting application control field data*)
TO(*ending application control field data*)
ACFIELD5(*starting application control field data*)
TO(*ending application control field data*)
BATCH(*translated transaction batch ID*)
DICTIONARY(*Dictionary name*)
DIR(*Direction: 'S' (Send/Source) and 'T' (Receive/Target)*)
DOCUMENT(*Document name*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IMAGE(*write image data record*)
NETID(*network ID*)
STDTRID(*EDI standard transaction set ID*)
SYNTAX(*syntax type: 'D'(data format) or 'X'(XML)*)
TPNICKN(*trading partner nickname*)
TRXDATE(*starting transaction date*)
TO(*ending transaction date*)
DOCSTAT(*Transaction store status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
DOCREC(*write document data record*)
EXTEND(*write the extended document record*)
CONCATENATE(*concatenate extract data*)

Notes:

1. ACFIELD will be interpreted as ACFIELD1.
2. IMAGE is not a new keyword. It is supported in the same way as it is with the current keywords of this command.
3. If DOCREC(N) is used, no data records or image records are returned.

Example

Retrieve the document information for all documents with batch ID of **ABCDE** and place the results in the EDIQUERY file.

```
PERFORM DOCUMENT DATA EXTRACT SELECTING DOCREC(Y) EXTEND(Y) WHERE BATCH(ABCDE)
```

ENVELOPE command

This command takes the EDI transactions from the Document Store that are available for enveloping, envelopes them, and places the results in an envelope file. The envelope file is the TD queue specified in the network profile member or a file specified in the command. Your selection criteria determine which transactions go into the interchange envelope. The enveloper sorts the transactions to create the fewest number of functional groups and interchange envelopes. For more information about envelope processing, see the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01

Syntax

ENVELOPE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
BATCH(*translated transaction batch ID*)
ENVPRBREAK(*start new envelope*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FILEID(*processing file ddname*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
FORMAT(*data format ID*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IACCESS(*IEXIT access*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
ITPBREAK(*new interchange envelope*)
ITYPE(*IEXIT program type*)
NETID(*network ID*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
RECOVERY(*recovery unit of work*)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
STDTRID(*EDI standard transaction set ID*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
VERIFY(*verify transaction status*)

XML(*XML required*)
XMLDICT(*XML dictionary address*)
XMLDTDS(*XML DTD path*)

Example 1

Envelope all the EDI documents in the Document Store that are associated with data format ID **PO850** or **CORP861**, and with network **IINR41**. The NETID keyword is included in both WHERE clauses to make sure that only **IINR41** transactions are selected. (Each WHERE clause is a separate set of selection criteria.

```
PERFORM ENVELOPE
    WHERE FORMAT(PO850) NETID(IINR41)
    WHERE FORMAT(CORP861) NETID(IINR41)
```

Example 2

Envelope all the EDI documents in the Document Store that are associated with data format ID **PO850** or **CORP861**, and with network **IINR41**. Place the results in file **NETQUEUE**. The NETID keyword is included in both WHERE clauses to make sure that only **IINR41** transactions are selected. (Each WHERE clause is a separate set of selection criteria.)

```
PERFORM ENVELOPE
    WHERE FORMAT(PO850) NETID(IINR41) FILEID(NETQUEUE)
    WHERE FORMAT(CORP861) NETID(IINR41)
```

ENVELOPE AND SEND command

This command combines the functions of the ENVELOPE and SEND commands, It takes EDI transactions from the Document Store, envelopes them, and places the results in a ddname file specified in the `Transaction data queue` field of the network profile member or in a file specified by the command. Selection criteria determine which transactions go into the interchange envelope. The enveloper sorts the transactions to create the fewest possible number of functional groups and interchange envelopes. WebSphere Data Interchange then sends the enveloped transactions to the network. This composite command performs faster than the separate commands

For more information on determining the number of groups and envelopes needed, see the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

Syntax

ENVELOPE AND SEND
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
BATCH(*translated transaction batch ID*)
ENVPRBREAK(*start new envelope*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FILEID(*processing file ddname*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
FORMAT(*data format ID*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IACCESS(*IEXIT access*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
ITPBREAK(*new interchange envelope*)
ITYPE(*IEXIT program type*)
NETID(*network ID*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
RECOVERY(*recovery unit of work*)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SEQNUM(*increment network profile member numbers*)
SETCC(*condition codes*)
STDTRID(*EDI standard transaction set ID*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)

TRERLVL(*maximum translation error level*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
VERIFY(*verify transaction status*)

Example 1

Envelope and send all documents in the Document Store that have a destination of trading partner **PISCES**.

```
PERFORM ENVELOPE AND SEND
    WHERE REQID(IINREQ) TPNICKN(PISCES)
```

Example 2

Envelope and send all documents in the Document Store with batch ID **121401**.

```
PERFORM ENVELOPE AND SEND
    WHERE REQID(IINREQ) BATCH(121401)
```

Example 3

Envelope and send all documents in the Document Store with batch IDs **INB11214** or **GEIS1214**.

```
PERFORM ENVELOPE AND SEND
    WHERE REQID(INB1REQ) BATCH(INB11214)
    WHERE REQID(GEISREQ) BATCH(GEIS1214)
```

Note: These transactions will be sent using different networks.

ENVELOPE DATA EXTRACT command

This command extracts detailed information about enveloped transactions, sorted by trading partner nickname, direction, interchange control number, or receiver ID.

You can use this command to create report data to:

- Report on the number of purchase orders sent in a given period of time
- Report on the total number of bytes sent in a given period of time (this can be useful for charging back costs to other departments based on their EDI usage)
- Create a customized functional acknowledgment tracking report by application or by department; for example, an exception report on purchase orders sent more than 2 days ago that have not been acknowledged
- Create an exception report flagging missing control numbers for inbound envelopes

This command can also be used for functions other than reporting, such as:

- Archiving Document Store data
- Loading status data directly into the application by application key; for example, the status for each invoice sent can be loaded into the billing system by invoice number

To extract detailed information about transactions, use the TRANSACTION DATA EXTRACT command (see 111).

For information about wildcard usage, see Table 2 on page 10.

Syntax

ENVELOPE DATA EXTRACT
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLICATION(*write application data record*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
CONCATENATE(*concatenate extract data*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GROUP(*write group data record*)

GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
IMAGE(*write image data record*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTERCHANGE(*write interchange data record*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
RECEIVEACKDATA(*write detailed acknowledgment data*)
RECEIVEACKIMAGE(*write receive acknowledgment record*)
SENDACKDATA(*write detailed acknowledgment data*)
SENDACKIMAGE(*write acknowledgment record*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRANSACTION(*write transaction data record*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
USERPGM(*user program*)

Example 1

The EDI users in Company XYZ want to view the status of their EDI documents for the last month. Specifically, the Purchasing department would like to look up purchase orders, by purchase order number, to determine the daily status. The report must contain:

- Document type (purchase order, invoice, and so on)
- Control numbers
- Date and time
- Trading partner nickname
- Reference number (purchase order number, invoice number, and so on)
- Transaction status

The programmers at Company XYZ have decided to use a ENVELOPE DATA EXTRACT command to collect the necessary information from WebSphere Data Interchange . They have written a COBOL program, which they run daily, that will format the records into a customized report. Users will read this customized report, searching on the reference numbers to easily find the status they are interested in.

```
PERFORM ENVELOPE DATA EXTRACT
    SELECTING CONCATENATE(Y) INTERCHANGE(Y) GROUP(Y) TRANSACTION(Y)
    WHERE HANDLE(20010201) TO(20010328)
```

Figure 1 shows an example of a customized report.

```
                                XYZ COMPANY

        CONTROL      SET                          TRADING  PO # or
TYPE    NUMBER       NUMBER     DATE     TIME    NICKNAME  INVOICE # STATUS
---- ----------- ----------- -------- -------- --------- -----------  ---------
PO   00000000375 00000003386 03/02/01 06:02:93 BOBS      PO0100332201  ENVELOPED
PO   00000000383 00000003394 03/01/01 11:12:53 BOBS4790  PO0100222210  ENVELOPED
PO   00000000388 00000003399 03/02/01 11:33:06 BOBS4790  PO0100232115  ENVELOPED
PO   00000000389 00000003400 03/01/01 06:52:23 BOBS4790                 ENVELOPED
PO   00000000377 00000003388 03/01/01 11:41:23 BOBS      PO0100123130  ENVELOPED
PO   00000000381 00000003392 03/01/01 06:22:56 BOBS4790  PO0121312322  ENVELOPED
IN   00000000145 00000000383 03/02/01 12:53:42 DLGLEVEL  IN94410       RECEIVED
IN   00000000144 00000000382 03/02/01 09:12:15 DLGLEVEL  IN95443       RECEIVED
PO   00010000130 00000004853 03/02/01 09:43:02 DLGOCCUR  IN76445       ENVELOPED
PO   00010000130 00000004854 03/01/01 07:25:16 DLGOCCUR  IN76835       ENVELOPED
PO   00010000130 00000004855 03/01/01 11:51:06 DLGOCCUR  IN76337       ENVELOPED
PO   00010000130 00000004856 03/02/01 06:34:29 DLGOCCUR  IN76838       ENVELOPED
```

Figure 1. Example of a customized report produced by the ENVELOPE DATA EXTRACT command

Example 2

Retrieve the interchange, group, and transaction information for all transactions in a particular envelope with trading partner name **PISCES** and interchange control number **000008888**. Place the results in the EDIQUERY file.

```
PERFORM ENVELOPE DATA EXTRACT
    SELECTING INTERCHANGE(Y) GROUP(Y) TRANSACTION(Y)
    WHERE TPNICKN(PISCES) INTCTLNO(000008888) DIR(S)
```

Example 3

Retrieve the interchange, group, and transaction information for all envelopes with an application sender ID of **SERVO**. Include application and image information.

```
PERFORM ENVELOPE DATA EXTRACT
    SELECTING INTERCHANGE(Y) GROUP(Y) TRANSACTION(Y) APPLICATION(Y)
     IMAGE(Y)
    WHERE APPSNDID(SERVO) DIR(S)
```

EXPORT command

This command uses the control file to extract setup information from WebSphere Data Interchange and write records to one or more files. (“Export/Import control file (CTLFILE)” on page 214 for ddname*s*.) You can request a particular file record format with the `EIFORMAT` keyword. If the Export/Import file is empty, the requested format is used. If the Export/Import file is not empty, the format of the data already in the file is used.

Syntax

EXPORT
ASSOBJ(*associated objects*.)
CTLFILE(*control file ddname*)
CTLTYPE(*control file type*)
EIFORMAT(*requested export file record format*)
IFCC(*override condition codes*)
KEYID(*ID*)
MBRNAME(*member name*)
NOMSG(*print extraneous messages*)
SETCC(*condition codes*)

Example

Export the information in fixed format using a control file named EXOUT.

```
PERFORM EXPORT
    WHERE CTLFILE(EXOUT) EIFORMAT(FIXED)
```

GLB DUMP command

This command is available only in the CICS environment, and is not required for normal processing. However, this command can be used to gather information required by WebSphere Data Interchange support personnel when debugging problems related to the persistent environment. The dump is written to the file associated with ddname EDIGDMP1 in the CICS startup JCL.

Syntax

GLB DUMP
IFCC(*override condition codes*)
LEVEL(dump or trace level)
RESET(*reset output file*)
SETCC(*condition codes*)

Example 1

Open EDIGDMP1 for output and generate a complete persistent environment dump.

```
PERFORM GLB DUMP
    WHERE LEVEL(1) RESET(Y)
```

Example 2

Open EDIGDMP1 for append and generate a persistent environment dump of just the data area.

```
PERFORM GLB DUMP
    WHERE LEVEL(12)
```

GLB TRACE command

This command is available only in the CICS environment and is not required for normal processing. However, you can use this command to gather information required by WebSphere Data Interchange support personnel when debugging problems related to the persistent environment. The trace is written to the file associated with ddname EDIGTRC1 in the CICS startup JCL.

Syntax

GLB TRACE
IFCC(*override condition codes*)
LEVEL(dump or trace level)
RESET(*reset output file*)
SETCC(*condition codes*)

Example 1

Open EDIGTRC1 for output and start a persistent environment trace for as much detail as possible.

```
PERFORM GLB TRACE
    WHERE LEVEL(3) RESET(Y)
```

Example 2

End the persistent environment trace.

```
PERFORM GLB TRACE
    WHERE LEVEL(0)
```

HOLD command

This command puts a transaction into held status. While in held status, no actions are permitted against the transaction that would change the status of the transaction. Nor is it purged automatically when its storage time expires. If the transaction is one of a related group, all transactions in the group are placed in held status. The RELEASE command restores held transactions to their former status or, if their storage time expired during the hold period, changes them to STORE-TIME-EXPIRED status.

The HOLD, PURGE, RELEASE, and UNPURGE commands share a common syntax.

Syntax

HOLD
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(network transaction status)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)

TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Place in hold status all EDI documents destined for trading partner **PISCES** that were translated on December 14, 2001.

```
PERFORM HOLD
    WHERE TPNICKN(PISCES) TRXDATE(01/12/14) DIR(S)
```

IMPORT command

This command uses the control file to read a file or group of files for importing setup information.

Syntax

IMPORT
ACTUSAGE(*activate imported usage*)
CTLFILE(*control file ddname*)
CTLTYPE(*control file type*)
DUPCHECK(*perform duplicate checks*)
IFCC(*override condition codes*)
KEYID*ID*
MBRNAME*member name*
REPLACE*replace the same named object*
NOMSG(*print extraneous messages*)
SETCC(*condition codes*)
USAGETID*map name*
WRTCTLNO(*import trading partner control number*)

Example

Import all records specified by the batch control file in TS queue **INCTL**.

```
PERFORM IMPORT
    WHERE CTLFILE(INCTL) CTLTYPE(TS)
```

LOAD LOG ENTRIES command

This command copies the selected HOLDFILE records back into the event log table. In a DB2 environment, you can restore deleted records to the event log by specifying the `ARCHIVEFILE` value from the UNLOAD LOG ENTRIES command as the value for the `HOLDFILE` keyword on this command.

Syntax

LOAD LOG ENTRIES
APPLID(*application ID*)
HOLDFILE(*event log hold file name*)
HOLDTYPE(*event log hold file type*)
IFCC(*override condition codes*)
LOGAEID(*starting event log associated entry ID*)
TO(*ending event log associated entry ID*)
LOGDATE(*starting event log date*) TO(*ending event log date*)
LOGFORM(*starting event log format ID*) TO(*ending event log format ID*)
LOGTIME(*starting event log time*) TO(*ending event log time*)
LOGUSER(*starting event log user ID*) TO(*ending event log user ID*)
NEWAPPLID(*new application ID*)
SETCC(*condition codes*)

Example

Copy the held records for **EDIFFS** with a log date of **12/14/01** back into the event log table.

```
PERFORM LOAD LOG ENTRIES
    WHERE APPLID(EDIFFS) LOGDATE(12/14/01)
```

NETWORK ACTIVITY DATA EXTRACT command

This command collects data about the network activity of the requestors defined in the mailbox (requestor) profile and generates reports based on your criteria. The information collected includes the total number of envelopes or characters sent by each requestor for any given day or range of days.

You can use this command to create reports to:

- Reconcile network charges
- List network usage by user ID
- List heaviest network users
- List inactive requestors

For information about using wild cards, see Table 2 on page 10.

Syntax

NETWORK ACTIVITY DATA EXTRACT
ACCTID(*network account ID*)
DAYS(*starting date*) TO(*ending date*)
DYNSQL(*use dynamic SQL*)
IFCC(*override condition codes*)
NETID(*network ID*)
NETNAME(*network name*)
REQID(*requestor ID*)
SETCC(*condition codes*)
USERID(*network user ID*)
USERPGM(*user program*)

Example 1

Your network charges jumped significantly last month and you want to understand which of your applications has increased its network usage. To do this, you need a report that shows the:

- Account ID
- User ID
- Number of envelopes exchanged, both sent and received
- Number of characters exchanged, both sent and received

```
PERFORM NETWORK ACTIVITY DATA EXTRACT
    WHERE DAYS(01/08/01) TO(01/08/31)
```

The data resulting from this command was written to file EDIQUERY, sorted by a local sort utility, and used as input to a user-written program that created the report shown in Figure 2 on page 48.

```
                    Network Traffic Activity Report              Date: 01/09/20
                                                                 Time: 12:31:02

 Interchange    Total
 Month  Network Name  Net ID   Account ID  User ID   S/R  Envelopes    Characters
 -----  ------------  -------  ----------  --------  ---  -----------  ----------
 08/01  AT&T Global   IINR41   XTEV        PURCHASE   S    7776           9998376
        Network                                      R    4356           2333157
                                           ACCTPAYB   S    1526           8867837
                                                      R     662            333457
                                                           ----------  ----------
                                                       14320          21532827
                             XTEW        RECEIVNG   S     856             82251
                                                      R    2856            434457
                                           TRAFFIC    R     770             66615
                                                           ---------   ----------
                                                        4482            583323
                    IINCICS XTEX         PURCHASE   S    4476           2648376
                                                      R     756            453457
                                           PRODPO     S    6625            783890
                                                      R     559            467457
                                                      ----------        ----------
                                                       12416          11408192
```

Figure 2. Example of a network traffic activity report

Example 2

Extract information to reconcile your EDI traffic through a particular account (**CMPY**) and user ID (**USER21**) on the network for the month of May during 2001. Include a list of the requestors that use that account/user ID on the network, and the number of messages sent and received by each requestor (only the traffic through WebSphere Data Interchange for this account and user ID).

```
PERFORM NETWORK ACTIVITY DATA EXTRACT
    WHERE ACCTID(CMPY) USERID(USER21) DAYS(01/05/01) TO(01/05/31)
```

PRINT ACKNOWLEDGMENT IMAGE command

This command writes an image to the RPTFILE of the functional acknowledgments for the transactions that match your selection criteria. The image does not include the envelope segments.

Syntax

PRINT ACKNOWLEDGMENT IMAGE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MERGED(*merge transaction image with functional acknowledgment*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SEGMENTED(*print segment on new line*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)

TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Print functional acknowledgment images for all EDI documents with batch ID **121401**.

```
PERFORM PRINT ACKNOWLEDGMENT IMAGE
    WHERE BATCH(121401)
```

Figure 3 shows an example of the Functional Acknowledgment Image report.

```
TF13       Functional Acknowledgment Image        Date:  01/12/14
                                                  Time:  12:12:12

Trading partner nickname:  PISCES
Transaction handle  . . :  20011214101533000001
Transaction status  . . :  Transaction accepted


AK1*IN*40088!AK2*810*000048118!AK5*A!AK9*A*1*1*1!
```

Figure 3. Example of a Functional Acknowledgement Image report

PRINT ACTIVITY SUMMARY command

This command creates a summary of activity for the inbound and outbound transactions that match your selection criteria. The summary is written to RPTFILE.

Syntax

PRINT ACTIVITY SUMMARY
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MERGED(*merge transaction image with functional acknowledgment*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SEGMENTED(*print segment on new line*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)

TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Print the transaction Activity Summary Report for all EDI documents entering the Document Store between November 29 and November 30, 2001.

```
PERFORM PRINT ACTIVITY SUMMARY
    WHERE TRXDATE(01/11/29) TO(01/11/30)
```

Note: This example is only valid when the date mask in the language profile (LANGPROF) is &Y/&M/&D.

Figure 4 shows an example of the Activity Summary Report.

```
TF11                Activity Summary Report                Date: 01/12/14
                                                           Time: 12:12:12

                 Outbound Transactions            Count
                 Selected transactions . . . . :   1940
                   Translation . . . . . . . . :   1940
                     Acceptably translated . . :   1930
                     Unacceptably translated . :     10
                   Enveloping  . . . . . . . . :   1000
                     Enveloped . . . . . . . . :    900
                     Enveloping errors . . . . :    100
                   Send requests . . . . . . . :    900
                     Sent  . . . . . . . . . . :    800
                       Pending functional ack  :     10
                       Pending network ack . . :     10
                     Not sent  . . . . . . . . :    100
                       Error on request to send:     73
                       Network error . . . . . :     27
                   Detached  . . . . . . . . . :      0


                 Inbound Transactions             Count

                 Selected transactions . . . . :   1000
                   Acceptably translated . . . :    800
                   Unacceptably translated . . :     10
                   Not yet translated  . . . . :    190
                   Detached  . . . . . . . . . :      0
```

Figure 4. Example of an Activity Summary Report

PRINT EVENT LOG command

This command writes an image of event log entries to RPTFILE for the transactions that match your selection criteria.

Syntax

PRINT EVENT LOG
ACFIELD(*application control field data*)
APPLICATION(*write application data record*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
CONCATENATE(*concatenate extract data*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*) TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GROUP(*write group data record*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
IMAGE(*write image data record*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTERCHANGE(*write interchange data record*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
RECEIVEACKDATA(*write detailed acknowledgment data*)
RECEIVEACKIMAGE(*write receive acknowledgment record*)
SENDACKDATA(*write detailed acknowledgment data*)
SENDACKIMAGE(*write acknowledgment record*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)

TRANSACTION(*write transaction data record*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
USERPGM(*user program*)

Example

Print log entries for all EDI documents that encountered errors during send processing on December 12, 2001 for trading partner **ABMETAL1**.

```
PERFORM PRINT EVENT LOG
    WHERE TRXSTAT(42) TRXDATE(01/12/14) TPID(ABMETAL1)
    WHERE TRXSTAT(43) TRXDATE(01/12/14) TPID(ABMETAL1)
```

Figure 5 on page 55 shows an example of the Events Logging Report.

```
                        EVENTS LOGGING REPORT

                        REQUESTOR . . . . : USER21

                        DATE  . . . . . . : 01/12/14

                        TIME  . . . . . . : 08:37:36

                      Selection Criteria for Log Report

                      Application ID . . . . :  EDIFFS
                      Date . . . . . . . . . :  01/07/21   To  01/07/28
                      Time . . . . . . . . . :
                      User ID  . . . . . . . :
                      Format ID  . . . . . . :
                      Associated Entry ID  . :  20010721164325210000
   EVENTS LOGGING REPORT  DATE:  01/12/14      TIME:  08:37:36      Page  1

 Date . . . . . . . . . . : 01/07/21
 Time . . . . . . . . . . : 16:43:26
 Entry ID . . . . . . . . : E00000000000000000000000000000200107211643262130000000
 Associated Entry ID  . : 20010721164325210000
 User ID  . . . . . . . . : USER21
 Job ID . . . . . . . . . : EDITR
 Application ID . . . . : EDIFFS
 Format ID  . . . . . . . : $$STD-'++:.?
 Data . . . . . . . . . . : UNB+UNOA:1+GBLLD.LLD324+GBLLD.LLD053:ZZ+930602:1646+
40000092++PLACIN++1++1'UNH+40000200+LIMUWM:1:2:LI'RFF+UMR+B9999LIMUWMUMR1'RFF+TR+B
9999LIMUWMTR1'RFF+BQR+B9999LIMUWMBQR1' NAD+BK+9999'NAD+BU+LL'NAD+CS+10
AEC'GIS+02:04'NAD+IV+ LINE 1:LL'GIS+01:26'GIS+01:15'GIS+Y:16'
RFF+LIR+0001'RFF+CS1+COSLINE 1'RFF+CS1+COSLINE
1'RFF+LIN+001'PCD+001:1.0000001'PCD+002:2.0000002'
PCD+003:3.0000003'PCD+004:4.0000004' PCD+005:5.0000005'L01+034+UNDERWRITER MESSAGE
LINE 1  TE'L02+1::LINE ONE OF THE INVITEDITER LINE 1  TEXT+3::LINE TWO OF THE
INVITEDITER LINE 1  TEXT+::LINE THREE  THE INVITEDITER LINE 1  TEXT'NAD+IV+
LINE2:LL'GIS+02:26'GIS+01:15'GIS+Y:16'RFF+LIR+0002'FF+CS1+COSLINE
2'RFF+CS1+COSLINE 2'RFF+LIN+002'PCD+001:1.0000001'PCD+002:2.0000002'
PCD+003:3.0000003' PCD+004:4.0000004'PCD+005:5.0000005'L01+034+UNDERWRITER MESSAGE
LINE 2  TE' L02+1::LINE ONE OF THE INVITEDITER LINE 2  TEXT+3::LINE TWO OF THE
INVITEDITER LINE 2  TEXT+::LINE THREE  THE INVITEDITER LINE 2
TEXT'UNT+39+40000200'UNZ+1+40000092'
The number of records printed from the input file was 2
Printing has terminated. Return code is 0; extended return code is 0.
```

Figure 5. Example of an Events Logging Report

PRINT STATUS SUMMARY command

This command extracts status information about the transactions that match your selection criteria and writes the information to RPTFILE. This report shows transaction status (translated, enveloped), network status (delivered, purged), and store status (active, held, marked for purging). An **R** (related) after the transaction handle indicates the transaction is a part of a bundle.

Status summaries are not available for online viewing.

Syntax

PRINT STATUS SUMMARY
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MERGED(*merge transaction image with functional acknowledgment*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SEGMENTED(*print segment on new line*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)

TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Print the status summary for all EDI documents that were translated on December 14, 2001.

```
PERFORM PRINT STATUS SUMMARY
    WHERE TRXDATE(01/12/14)
```

Figure 6 shows an example of the Status Summary Report for Outbound Transactions.

```
TF80                                    Status Summary Report for              Date: 01/12/14
                                          Outbound Transactions                Time: 12:12:12

Transaction Handle  Trading Partner Nickname   Data Format ID    Transaction Status    Store Status
    Date Enveloped    Interchange Cntrl No   Network Status        Group Control No     Func Ack Status
200112140930120000   Partner111111111          Format1111111111  Transaction accepted  Purge Requested
    01/12/14          12121212121212         Accepted by network   11111111111111       Received

200112121045330001   Partner111111111          Format3333333333  Send translate error  Active

200112141033110000   Partner222222222          Format2222222222  Transaction accepted  Active
    01/12/14          22222222222222         Accepted by network   11111111111111       Received

200112141033110004 R Partner222222222          Format3333333333  Transaction accepted  Active
    01/12/14          22222222222222         Accepted by network   22222222222222       Received

200112141033110003   Partner333333333          Format1111111111  Transaction accepted  Purge Requested
    01/12/14          33333333333333         Recall request error  11111111111111       Received
    01/12/14          33333333333333         Not sent - net error  22222222222222
```

Figure 6. Example of the Status Summary Report for Outbound Transactions

The same report format is used for Status Summary reports for Inbound Transactions, shown in Figure 7 on page 58.

```
TF80                                        Status Summary Report for              Date:  01/12/14
                                               Inbound Transactions                Time:  12:12:12

Transaction Handle    Trading Partner Nickname    Standard Trans ID  Transaction Status   Store Status
              Date Translated   Data Format ID       Translation Status
200112141344270000    Partner222222222            12121212           Receive translated   Active
              01/12/14          Format1212121212    Acceptable

200112141010100001    Partner222222222            12345678           Receive translated   Active
              01/12/14          Format3333333333    Acceptable
              01/12/14          Format3333333333    Unacceptable
```

Figure 7. Example of the Status Summary Report for Inbound Transactions

PRINT STATUS SUMMARY2 command

This command extracts the same information as the PRINT STATUS SUMMARY command but adds a line containing the application control number and the internal trading partner ID for each transaction. An **R** (related) after the transaction handle indicates the transaction is part of a bundle.

Status summaries are not available for online viewing.

Syntax

PRINT STATUS SUMMARY2
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MERGED(*merge transaction image with functional acknowledgment*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SEGMENTED(*print segment on new line*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)

TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Print transaction images for all EDI documents with receive translation errors.

```
PERFORM PRINT STATUS SUMMARY2
    WHERE TRXSTAT(73)
```

Figure 8 shows an example of the Selection Criteria for Status Summary Report.

```
1                      Selection Criteria for                Date: 01/12/14
                        Status Summary Report                Time: 14:25:44

SELECTION CRITERIA      1

Transaction handle. . . . . . . . : 20010323000000000000  TO  20010323240000000000
1TF80                    Status Summary Report for                Date:  01/12/14
                           Outbound Transactions                  Time:  14:25:44

Transaction Handle  Trading Partner Nickname  Data Format ID  Transaction Status Store Status
       Application Control Number            Internal Trading Partner ID
Date Enveloped  Interchange Cntrl No  Network Status    Group Control No  Func Ack Status

200103230921579000  BOBS                     BOBS8004833     ENVELOPED   ACTIVE
                                             1111122222
01/12/14        00000000000418        ENVELOPED         00000000000399    PENDING

20011214116562000   BOBS                     BOBSPORDER      ENVELOPED   ACTIVE
        BOBS      12345678                   BOBS
01/12/14        00000000000419        ENVELOPED         00000000000400    NOT REQUESTED
```

Figure 8. Example of the Selection Criteria for Status Summary Report

PRINT TRANSACTION DETAILS command

This command extracts detailed information about the transactions that match your selection criteria and writes the information to RPTFILE.

Syntax

PRINT TRANSACTION DETAILS
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MERGED(*merge transaction image with functional acknowledgment*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SEGMENTED(*print segment on new line*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)

TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Print transaction details for all EDI documents that were placed in the Document Store on December 14, 2001.

```
PERFORM PRINT TRANSACTION DETAILS
    WHERE TRXDATE(01/12/14)
```

Figure 9 shows an example of the Transaction Details report.

```
TF73                Transaction Details                Date:  01/12/14
                                                       Time:  12:12:12


Transaction handle . . . . . :  20011214101533000001
Trading partner nickname . . . . . :  PISCES
Internal trading partner ID  . . . . . :  23456789012345678901234567890123 45
Direction  . . . . . :  SEND
Data format ID . . . . . :  POSEND
Application control number . . . . . :  12345678901234567890123456789012345
Interchange control number . . . . . :  12345678901234
Group control number . . . . . :  12345678901234
Transaction control number . . . . . :  12345678901234
Transaction status . . . . . :  Translated
Added to store . . . . . :  01/12/14-10:29:48
Store status   . . . . . :  Active
Delivered to application . . . . . :
Translation error level  . . . . . :  1
Translation  . . . . . . . . . . . . :  Acceptable
Network acknowledgment requested . . . :  D
Network status . . . . . :
Functional acknowledgment  . . . . . :
Functional ack date  . . . . . :
Application ID . . . . . :  EDIMP
Batch ID . . . . . :  BATCH123
Earliest purge date  . . . . . :  01/12/14
Earliest envelope date . . . . . :  01/12/14
Enveloped/deenveloped  . . . . . :
Sent . . . . . :
Envelope profile member  . . . . . :  X12V2R2
Standard ID  . . . . . :  X12V2R2
Standard version . . . . . :  V2
Standard level . . . . . :  R2
Standard transaction ID  . . . . . :  850
Network ID . . . . . :  IN
Usage indicator  . . . . . :  P
Total segment count  . . . . . :  22
Transaction size (bytes) . . . . . :  742
```

Figure 9. Example of the Transaction Details report

PRINT TRANSACTION IMAGE command

This command writes an image of the transactions that match your selection criteria to RPTFILE. The images do not include envelope or security segments.

Syntax

PRINT TRANSACTION IMAGE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MERGED(*merge transaction image with functional acknowledgment*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SEGMENTED(*print segment on new line*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)

TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Print transaction images for all EDI documents with receive translation errors.

```
PERFORM PRINT TRANSACTION IMAGE
    WHERE TRXSTAT(73)
```

Figure 10 shows an example of the Transaction Image report.

```
TF75                      Transaction Image                  Date:  01/12/14
                                                             Time:  12:12:12

Trading partner nickname:  PISCES
Transaction handle  . . :  20011214101533000001
Transaction status  . . :  Receive trans error

SEG1*dATA:dATAN**dANTA:dANTAN:-99:-99!SEG2*123456:-1.00:-100*12.3456:-
1.00:010920:123000:-10000!SEG3*123.456:99:001128:083
0:990*1234.56:-1:010920:1259:-1*1234.56:-1.0000:010920:125900:-
100!SEG4*15:1.00:001128:1259:1000*15:-1.0:001128:1259:99*C1

C2:D1D2D3D4D5:1.001128:1259:1000!SEG5*001128*1259!TOTALS*210.000000*96.000000*3
28.000000*196.000000*32.000000*8.000000!
```

Figure 10. Example of the Transaction Image report

PROCESS command

This is a generic command that tells WebSphere Data Interchange to execute the commands in a service profile. It enables you to author and maintain your commands (along with their data sources and targets) in the client tool, rather than in the programs that invoke WebSphere Data Interchange . You name and specify the services you want WebSphere Data Interchange to provide to your applications by creating service profiles. You then have your applications call WebSphere Data Interchange and submit a **PROCESS** command naming the services you created. If you decide to change the command a service invokes, or specify different options for the command, you simply update the service profile. You do not need to update or recompile your application programs. Complex services can require multiples commands and in addition, different input messages can require different processing. The **PROCESS** command invokes the ″command chaining″ feature of WebSphere Data Interchange that will chain together multiple service profiles based upon the destination of the output data from the previous command in the chain.

Note: Each ″Command Chaining″ that a **PROCESS** command invokes is considered as a single translation session. Therefore, all Global session variables and accumulators get reset for each of the **PROCESS** command.

The **PROCESS** and **RECEIVE AND PROCESS** commands are the only commands that invoke the command chaining feature in WebSphere Data Interchange .

Syntax

PROCESS
FILEID(*logical name of the source data*)

Example 1

XML purchase orders are sent from the purchasing system to WebSphere Data Interchange to be transformed into EDI and sent to the trading partner. A service profile called **XML_IN** is created in the WebSphere Data Interchange client tool. Each XML purchase order is written to a file whose logical name is set to **XML_IN** and then WebSphere Data Interchange is invoked with the following command:

```
PERFORM PROCESS
   WHERE FILEID(XML_IN)
```

Example 2

Vendor catalogs in EDI format are received from trading partners using FTP. The catalogs must be translated into the internal proprietary XML format and loaded into the internal application. A service profile called **EDI_IN** is created in the WebSphere Data Interchange client tool and each time a catalog is received, the logical name of the file received is set to **EDI_IN** and then WebSphere Data Interchange is invoked with the following command:

```
PERFORM PROCESS
   WHERE FILEID(EDI_IN)
```

PROCESS NETWORK ACKS command

This command processes acknowledgments that have already been received into a file. It is used internally by WebSphere Data Interchange when network acknowledgments are processed on a continuous receive basis. This command can also be used if network acknowledgments are received outside WebSphere Data Interchange control, but must still be applied to the Transaction Store.

Syntax

PROCESS NETWORK ACKS
ACKFILE(*network acknowledgment ddname*)
ACKTYPE(*acknowledgment file type*)

Example

Process a file allocated to ddname **PROCACKS** containing network acknowledgments associated with requestor ID **IINB41REQ**.

```
PERFORM PROCESS NETWORK ACKS
    WHERE REQID(IINB41REQ) ACKFILE(PROCACKS)
```

PURGE command

This command marks a transaction for purging from the Document Store but does not remove it. Transactions are marked for purging when old outbound transactions are still pending functional reconciliation. Transactions marked for purge are deleted from the store when you execute the REMOVE TRANSACTIONS command. For more information, see “REMOVE TRANSACTIONS command” on page 90. If the transaction is one of a related group, all transactions in the group are marked for purging. Only transactions with status of PURGE-USER REQUEST or PURGE-DATE EXPIRED are eligible for purging. Reconciled transactions are automatically marked PURGE-DATE EXPIRED after the purge internal has expired.

The default expiration date is 30 days from translation. You can change the default by specifying a PURGINT value on PERFORM statements that add transactions to the Document Store (for example, TRANSLATE AND ENVELOPE). See “PURGINT” on page 183 for applicable PERFORM commands.

The HOLD, PURGE, RELEASE, and UNPURGE commands share a common syntax.

Syntax

PURGE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)

SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Mark for purging all EDI documents that have been delivered to trading partner **PISCES** and accepted.

```
PERFORM PURGE
    WHERE TPNICKN(PISCES) TRXSTAT(61)
```

QUERY command

This command generates a list of transactions that meet the selection criteria you specify in the keywords. In the returned list, transactions are identified by their transaction handle (`Document Store ID`). WebSphere Data Interchange writes the list of handles to the specified file (ddname EDIQUERY for z/OS). Each item in the list is 30 bytes long and has the format shown in Table 5:

Table 5. Transaction handle format

Bytes	Handle format
1–10	Packed decimal
11–30	EBCDIC character

You can use the QUERY command to provide input for a user-written reporting program.

Note: In CICS, you can pass the file name and type to WebSphere Data Interchange as parameters.

Syntax

QUERY
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)

NETID(*network ID*)
NETSTAT(*network transaction status*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Return a list of transactions with batch ID **121401** and translation error level **2** (data element and segment errors).

```
PERFORM QUERY
    WHERE BATCH(121401) TRERLVL(2)
```

QUERY PROFILE command

You can query for information about defined profiles. The results are written in tagged, fixed, or native format to the file specified in the command. The default format is **FIXED**. The default output file name is **EDIQUERY**, and the default output file type in CICS is **TS**. If no member is specified in the command, the entire profile is queried for the requested information.

Syntax

QUERY PROFILE
ID(*DataInterchange profile name*)
IFCC(*override condition codes*)
MEMBER(*member profile name*)
OUTFILE(*output file name*)
OUTFORMAT(*output format*)
OUTTYPE(*output file type*)
SETCC(*condition codes*)

Example 1

Query all trading partner profile members and write results to **MYFILE** in fixed format.

```
PERFORM QUERY PROFILE
    WHERE ID(TPPROF) OUTFILE(MYFILE)
```

Example 2

Query mailbox (requestor) profile member **REQMEM** and write results to **MYFILE** in the original format.

```
PERFORM QUERY PROFILE
    WHERE ID(REQPROF) MEMBER(REQMEM) OUTFILE(MYFILE) OUTFORMAT(N)
```

RECEIVE command

This command receives EDI data from the network identified in the mailbox (requestor) profile member. It places the data in the receive file specified in the mailbox (requestor) profile or in the file specified in the command.

Syntax

RECEIVE
CLEARFILE(*clear specified file contents*)
ENVTYPE(*transaction envelope type*)
FILEID(processing file ddname)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
ONEMSG(*read only one MQ message*)
REQID(requestor ID)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example 1

Receive data for requestors defined in mailbox (requestor) profile members **DEPTA7F** and **DEPTB8R**. Place the data into the receive files specified by the profile members. Clear both files before the data is received. The data is in X12 format.

```
PERFORM RECEIVE
    WHERE REQID(DEPTA7F) CLEARFILE(Y)
    WHERE REQID(DEPTB8R) CLEARFILE(Y)
```

Example 2

In CICS, receive data for the requestor defined by mailbox (requestor) profile member **DEPTA7F**. Place the data in CICS temporary storage queue **RECVPO**. The data is in EDIFACT format.

```
PERFORM RECEIVE
    WHERE REQID(DEPTA7F) FILEID(RECVPO) ENVTYPE(E)
```

RECEIVE AND DEENVELOPE command

This command combines the functions of the RECEIVE and DEENVELOPE commands. It receives EDI data from the network indicated in the mailbox (requestor) profile member and places the data in the receive file specified by the mailbox (requestor) profile member or in the file specified in the command. It then removes the envelope segments and places the results in the Document Store. This composite command performs faster than the separate commands.

Syntax

RECEIVE AND DEENVELOPE
CLEARFILE(*clear specified file contents*)
DIERRFILTER(*initial error filter set*)
DUPENV(*process duplicate envelopes*)
ENVTYPE(*transaction envelope type*)
EXTENDC(*translate with extended C record format*)
FADELAY(*delay queuing functional acknowledgment*)
FILEID(*processing file ddname*)
FORCETEST(*force test usage*)
FUNACKFILE(*functional acknowledgment ddname*)
FUNACKREQ(*require functional acknowledgment envelope file*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
MRREQID(*management reporting requestor ID*)
MSGUCLASS(*override message user class*)
ONEMSG(*read only one MQ message*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RECOVERY(*recovery unit of work*)
REQID(requestor ID)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example 1

Receive all EDI documents for requestor **DEPTA7F** and deenvelope them. Place them in the Document Store. The documents are in X12 format.

```
PERFORM RECEIVE AND DEENVELOPE
    WHERE REQID(DEPTA7F)
```

Example 2

Receive all EDI documents for requestor **DEPTA7F**, deenvelope them. Place them in the Document Store. Set the purge interval for these documents at 20 days. The documents are in EDIFACT format.

```
PERFORM RECEIVE AND DEENVELOPE
    WHERE REQID(DEPTA7F) PURGINT(20) ENVTYPE(E)
```

RECEIVE AND PROCESS command

This is a combined command that tells WebSphere Data Interchange to do a **PERFORM RECEIVE** followed by a **PERFORM PROCESS**. (See “RECEIVE command” on page 72and “PROCESS command” on page 65 for more information).

The **RECEIVE AND PROCESS** command is most commonly used in trigger programs. For example, when a message arrives on a WebSphere MQ queue, the WebSphere Data Interchange Adapter for WebSphere MQ (a trigger program) invokes WebSphere Data Interchange and passes it a **RECEIVE AND PROCESS** command which causes WebSphere Data Interchange to receive the message off of the queue and process it as specified in the associated Service Profile. The valid keywords on this command are the keywords that are valid for a **PERFORM RECEIVE** command. (see “Syntax” on page 72 for more information).

Syntax

RECEIVE AND PROCESS
CLEARFILE(*clear specified file contents*)
ENVTYPE(*transaction envelope type*)
FILEID(*logical name of the source data*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
ONEMSG(*read only one MQ message*)
REQID(*requestor ID*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

XML purchase orders are routed from the purchasing system to WebSphere Data Interchange by WebSphere MQ Integrator on a WebSphere MQ queue named **XML_IN**. WebSphere Data Interchange job is to transform them into EDI and send to the appropriate trading partner. The customer has elected to write their own trigger program instead of using the WebSphere Data Interchange Adapter for WebSphere MQ since they want more control over the triggering process. A service profile called **SNDXMLPO** is created in the WebSphere Data Interchange client tool. Whenever a message arrives on the **XML_IN** queue, the trigger program invokes WebSphere Data Interchange with the following command:

```
PERFORM RECEIVE AND PROCESS
   WHERE REQID(XML_IN) FILEID(SNDXMLPO) CLEARFILE(Y)
```

RECEIVE AND SEND command

This command is valid only with MQSeries queues and combines the functions of the RECEIVE and SEND commands. Data can be received from an MQSeries queue, the data translated, and the translated output placed in a destination MQSeries queue.

Syntax

RECEIVE AND SEND
APPFILE(*application data file name*)
CLEARFILE(*clear specified file contents*)
ENVTYPE(*transaction envelope type*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
RAWDATA(*translate to raw data format*)
REQID(requestor ID)
SAPUPDT(*track SAP status*)
SCRIPT(*script ID*)
SEQNUM(*increment network profile member numbers*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

Receive raw data from MQSeries queue **MQTESTREQ1**. Place the data into the receive file **MQSM**.

```
PERFORM RECEIVE AND SEND
    WHERE REQID(MQTESTREQ1) RAWDATA(Y) APPFILE(MQSM)
```

RECEIVE AND TRANSLATE command

This command combines the functions of the RECEIVE, DEENVELOPE, and TRANSLATE TO APPLICATION commands. This composite command performs faster than the separate commands.

Note: In CICS, you can also deliver the data to a program or CICS transaction, as specified in the `Application file name` and `Application file type` fields of the data format.

Syntax

RECEIVE AND TRANSLATE
ASSERTLVL(*session assertion level*)
BATCHSET(*set transaction batch ID*)
CCEXCEPTION(*job-step condition code*)
CLEARFILE(*clear specified file contents*)
DIERRFILTER(*initial error filter set*)
DUPENV(*process duplicate envelopes*)
ENVTYPE(*transaction envelope type*)
EXTENDC(*translate with extended C record format*)
FADELAY(*delay queuing functional acknowledgment*)
FILEID(*processing file ddname*)
FORCETEST(*force test usage*)
FUNACKFILE(*functional acknowledgment ddname*)
FUNACKREQ(*require functional acknowledgment envelope file*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
ONEMSG(*read only one MQ message*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RAWDATA(*translate to raw data format*)
REQID(requestor ID)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example 1

Receive and translate all EDI documents for requestor **DEPTA7F**. Along with the transaction data, return all optional records produced during deenveloping and translation. The documents are in X12 format.

```
PERFORM RECEIVE AND TRANSLATE
    WHERE REQID(DEPTA7F) OPTRECS(IEGTQ)
```

Example 2

Receive and translate all EDI documents for requestor **DEPTA7F**. Return the transaction data in raw data format. The documents are in EDIFACT format.

```
PERFORM RECEIVE AND TRANSLATE
    WHERE REQID(DEPTA7F) RAWDATA(Y) ENVTYPE(E)
```

RECONSTRUCT command

This command takes information that has been saved in the Document Store and rebuilds an interchange just as it was sent or received (using the same control numbers). This command can be used if your trading partner lost an interchange you sent, and you must send the same interchange again. It can also be used to rebuild interchanges sent to you.

Syntax

RECONSTRUCT
DIR(*processing direction*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

You received and processed interchange control number **5** from your trading partner **MYTP**, but have since destroyed the sequential file containing the interchange. You would like to reconstruct this file so you can save it for the auditors. The command statement will reconstruct the interchange for trading partner **MYTP** with interchange receiver ID **123456789**, interchange control number **5**, and file name **AUDITOR**.

```
PERFORM RECONSTRUCT
    WHERE TPNICKN(MYTP) INTRECID(123456789) INTCTLNO(5) DIR(R)
    FILEID(AUDITOR)
```

RECONSTRUCT AND SEND command

This command combines the functions of the RECONSTRUCT and SEND commands. It rebuilds interchanges from data in the Document Store, and then sends those interchanges to the network. This composite command performs faster than the separate commands.

Syntax

RECONSTRUCT AND SEND
CLEARFILE(*clear specified file contents*)
DIR(*processing direction*)
ENVTYPE(*transaction envelope type*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MSGUCLASS(*override message user class*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
REQID(*requestor ID*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

Your trading partner **MYTP** called and indicated that interchange control number **5** is missing, and that they would like you to resend that interchange. The trading partner's DUNS number **123456789**is used as the interchange receiver ID value. The command statement will reconstruct the interchange for trading partner **MYTP** with interchange receiver ID **123456789** and interchange control number **5**.

```
PERFORM RECONSTRUCT AND SEND
    WHERE TPNICKN(MYTP) INTRECID(123456789) INTCTLNO(5) DIR(S)
    REQID(MYREQ)
```

RECVFILE command

This command receives application data files that do not need an interchange header or that do not contain EDI data. The data is not necessarily in any EDI standard format.

Syntax

RECVFILE
CLEARFILE(*clear specified file contents*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
REQID(*requestor ID*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

Receive data for requestors defined in mailbox (requestor) profile member **DEPTA7F**. Place the data into file **RCVFILE**. Clear **RCVFILE** before the data is received.

```
PERFORM RECVFILE
    WHERE REQID(DEPTA7F) FILEID(RCVFILE) CLEARFILE(Y)
```

RECVFILE AND SEND command

This command is valid only with MQSeries queues and combines the functions of the RECVFILE and SEND commands. Application data can be received from an MQSeries queue, the data translated, and the translated output placed in a destination MQSeries queue. The data is not necessarily in any EDI standard format.

Syntax

RECVFILE AND SEND
CLEARFILE(*clear specified file contents*)
ENVTYPE(*transaction envelope type*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SCRIPT(*script ID*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

Receive raw data from MQSeries queue **MQTESTREQ1**. Place the data into the receive file **QDATA**.

```
PERFORM RECVFILE AND SEND
    WHERE REQID(MQTESTREQ1) FILEID(QDATA) RAWDATA(Y)
```

REENVELOPE command

This command takes the EDI transactions from the Document Store that were previously enveloped, envelopes them again, and places the results in an envelope file. The envelope file is either the TD queue specified in the network profile member or a file specified in the command. Selection criteria determine which transactions go into the interchange envelope. The enveloper sorts the transactions to create the fewest number of functional groups and interchange envelopes. For more information about envelope processing, see the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

Syntax

REENVELOPE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIERRFILTER(*initial error filter set*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVPRBREAK(*start new envelope*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FILEID(*processing file ddname*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IACCESS(*IEXIT access*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
ITPBREAK(*new interchange envelope*)
ITYPE(*IEXIT program type*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
OPTRECS(*optional record type*)

OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
RECOVERY(*recovery unit of work*)
SAPUPDT(*track SAP status*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
STDTRID(*EDI standard transaction set ID*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
VERIFY(*verify transaction status*)

Example

Reenvelope the EDI document with the Document Store handle **20011214101533000001** and a status of **31**. Place the results in the network TD queue.

```
PERFORM REENVELOPE
    WHERE HANDLE(20011214101533000001) TRXSTAT(31)
```

REENVELOPE AND SEND command

This command provides the same functions as the ENVELOPE AND SEND command but for EDI data that was previously enveloped. You must supply the requestor ID of each network for which data is reenveloped. This composite command performs faster than the separate commands.

Syntax

REENVELOPE AND SEND
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
CLEARFILE(*clear specified file contents*)
DIERRFILTER(*initial error filter set*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVPRBREAK(*start new envelope*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FILEID(*processing file ddname*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IACCESS(*IEXIT access*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
ITPBREAK(*new interchange envelope*)
ITYPE(*IEXIT program type*)
MSGUCLASS(*override message user class*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name for SAP Status*)
OUTTYPE(*output data file type for SAP Status*)

PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
RECOVERY(*recovery unit of work*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SCRIPT(*script ID*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
STDTRID(*EDI standard transaction set ID*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TPNICKNESEND(*trading partner profile member*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
VERIFY(*verify transaction status*)

Example

Reenvelope the EDI document with transaction handle **20011212101533000001** and a status of **31**. Place the results in the network queue and send the data.

```
PERFORM REENVELOPE AND SEND
    WHERE HANDLE(20011212101533000001) TRXSTAT(31) REQID(MYREQ)
```

RELEASE command

This command restores a transaction in held status to its former status (or to PURGE-PENDING status if its store time expired during the hold period). If the transaction is one of a related group, all transactions in the group are released. WebSphere Data Interchange never automatically sets a transaction's store status to HELD; you must use the HOLD command.

The HOLD, PURGE, RELEASE, and UNPURGE commands share a common syntax.

Syntax

RELEASE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)

ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Release all EDI documents destined for trading partner **PISCES** that are in HELD status.

```
PERFORM RELEASE
    WHERE TPNICKN(PISCES) DIR(S) STSTAT(1)
```

REMOVE AUDIT TRAIL command

This command is valid only for the audit trail and can be a very quick way to remove audit trail entries. You can run this command as the first step of a multi-step maintenance process.

Syntax

REMOVE AUDIT TRAIL
LOGDATE(*starting date*) TO(*ending date*)
LOGUSER(*starting user ID*)

Example

Remove audit trail entries from audit trail table for a certain date, range, or userid:

```
PERFORM REMOVE AUDIT TRAIL
    WHERE LOGDATE(12/14/06)

PERFORM REMOVE AUDIT TRAIL
    WHERE LOGDATE(12/14/06) TO(12/17/06)

PERFORM REMOVE AUDIT TRAIL
    WHERE LOGUSER(USR0890)
```

REMOVE SUBMISSION command

This command allows the user to delete job submission print files.

Syntax

REMOVE SUBMISSION
LOGDATE(*starting date*) TO(*ending date*)
LOGUSER(*starting user ID*)

Example

Delete job submission print files for a certain date, range, or userid:

```
PERFORM REMOVE SUBMISSION
    WHERE LOGDATE(12/14/06)

PERFORM REMOVE SUBMISSION
    WHERE LOGDATE(12/14/06) TO(12/17/06)

PERFORM REMOVE SUBMISSION
    WHERE LOGUSER(USR0890)
```

REMOVE LOG ENTRIES command

This command is valid only for DB2 event logs and can be a very quick way to remove entries. You can run this command as the first step of a multi-step maintenance process.

If the `Archivefile` keyword is used, this command will work the same as the UNLOAD LOG ENTRIES command by copying selected entries to the archive file.

If the `Numdels` keyword is used, rows are deleted sequentially with COMMITs performed each time the number of rows specified in `Numdels` is reached. If the `Numdels` keyword is omitted, a single fully-qualified SQL DELETE statement is issued. (This can be very efficient, but can also quickly exhaust DB2 resources. Omit `Numdels` with caution.) To ensure concurrency, set the `NUMDELS keyword` to a relatively low value.

Syntax

REMOVE LOG ENTRIES
APPLID(*application ID*)
ARCHIVEFILE(*event log archive file name*)
ARCHIVETYPE(*event log archive file type*)
HOLDFILE(*event log hold file name*)
HOLDTYPE(*event log hold file type*)
IFCC(*override condition codes*)
LOGAEID(*starting event log associated entry ID*)
TO(*ending event log associated entry ID*)
LOGDATE(*starting event log date*) TO(*ending event log date*)
LOGFORM(*starting event log format ID*) TO(*ending event log format ID*)
LOGTIME(*starting event log time*) TO(*ending event log time*)
LOGUSER(*starting event log user ID*) TO(*ending event log user ID*)
NUMDELS(*number of database deletes before commit*)
SETCC(*condition codes*)

Example

Remove log entries from application log file for application EDIFFS dated December 14, 2001.

```
PERFORM REMOVE LOG ENTRIES
    WHERE APPLID(EDIFFS) LOGDATE(12/14/01)
```

REMOVE STATISTICS command

This command deletes outdated management reporting statistics. If you do not specify a date, tomorrow's date is used as the selection criteria. To reset the cumulative records to zero, use the RESET STATISTICS command.

Syntax

REMOVE STATISTICS
IFCC(*override condition codes*)
NUMDELS(*number of database deletes before commit*)
PRIORTO(*deletion end date*)
SETCC(*condition codes*)

Example

Remove any statistics over six months old. In addition, suppose you use the DB2 version of WebSphere Data Interchange and previously exceeded the maximum number of page locks when issuing a REMOVE STATISTICS command. To get rid of statistics over six months old, use the `PRIORTO` keyword with a date of six months ago. To reduce the accumulation of page locks by the REMOVE STATISTICS process, you can set the `Numdels` parameter to a low number, such as 25. This causes the REMOVE STATISTICS command to commit work and release page locks after every 25 deletes.

```
PERFORM REMOVE STATISTICS
    WHERE PRIORTO(*-180) NUMDELS(25)
```

Notes:

1. Running this command does not remove cumulative records.
2. To reset the Management Reporting cumulative statistics counts to zero, use the RESET STATISTICS command.

REMOVE TRANSACTIONS command

This command deletes transactions from the Document Store that have a status of either:

- PURGE-STORE TIME EXPIRED
- PURGE-USER REQUEST

You can protect transactions you want to keep with the HOLD or UNPURGE command. UNPURGE, however, protects only transactions with a status of PURGE-USER REQUEST. It does not protect those with a status of PURGE-STORE TIME EXPIRED. Once deleted, transactions are not recoverable.

The default expiration date is 30 days from translation. You can change the default by specifying a PURGINT value on PERFORM statements which add transactions to the Document Store (for example, TRANSLATE AND ENVELOPE). See “PURGINT” on page 183 for applicable PERFORM commands.

Note: Running the REMOVE TRANSACTIONS command with selection criteria specified in the WHERE clause can remove other entries from the EDIVTSTH table that do not match the selection criteria. These additional entries reported as being removed in the audit trail. These entries are removed because they have no direction assigned to them and are over 10 days old. “Orphan” entries like these are created when the first transaction of a translation process does not complete normally. These entries cannot be accessed in any way and are only shown when removed. They are removed whenever the REMOVE TRANSACTIONS command is executed.

The WHERE clause is optional for this command. If you do not specify a WHERE clause, all eligible transactions are removed from the Document Store.

After purging documents from the Document Store, you can use the Archive action to remove the event logs associated with these transactions.

This command does not delete ACTIVE or HELD transactions.

- To make ACTIVE transactions eligible for removal, they must first be marked for purge.
- To make HELD transactions eligible for removal, they must first be released from held status.

Syntax

REMOVE TRANSACTIONS
ACFIELD(*starting application control field data*)
TO(ending application control field data)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)

DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*) TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MAXRUNTIME(*maximum remove runtime minutes*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(network transaction status)
NUMDELS(*number of database deletes before commit*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STANDALONE(*operate DataInterchange only*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Delete all eligible transactions that are more than 30 days old. Only transactions that have a status of PURGE-USER REQUEST or PURGE-DATE EXPIRED are eligible.

```
PERFORM REMOVE TRANSACTIONS
    WHERE HANDLE(*-999) TO(*-30)
```

REPORT CONTINUOUS RECEIVE STATUS command

This command generates a continuous receive status report about the status of either a single continuous receive member or all continuous receives known to WebSphere Data Interchange and GXS Expedite/CICS. The output goes to the report file. The report file name and type can be specified in the Utility Control Information block. The default file name is RPTFILE and the default file type in CICS is **TS**.

Syntax

REPORT CONTINUOUS RECEIVE STATUS
IFCC(*override condition codes*)
MEMBER(*member profile name*)
SETCC(*condition codes*)

Example 1

Report all continuous receives.

```
PERFORM REPORT CONTINUOUS RECEIVE STATUS
```

Example 2

Report the status of continuous receive profile member CRMEM.

```
PERFORM REPORT CONTINUOUS RECEIVE STATUS
    WHERE MEMBER(CRMEM)
```

RESET STATISTICS command

This command resets the cumulative records to zero. Run this command after running the REMOVE STATISTICS command to restart the statistical history.

Syntax

RESET STATISTICS
IFCC(*override condition codes*)
NUMDELS(*number of database deletes before commit*)
PRIORTO(*deletion end date*)
SETCC(*condition codes*)

Example

Reset the cumulative record values recorded for the last 180 days to zero.

```
PERFORM RESET STATISTICS
    WHERE PRIORTO(*-180)
```

RESTART RECEIVE command

This command is used when receiving EDI data, and is valid only when both the following are true:

- The network supports restart
- You specify checkpoint-level recovery when initially receiving the data

If an error causes network processing to enter a restart situation while processing EDI data, this command can be used to restart and complete the receive. For more information on checkpoint recovery, see the *GXS Expedite Base/MVS Programming Guide*.

Notes:

1. The requestor ID must be the same value specified on the initial receive.
2. This command is not supported in the CICS environment.

Syntax

RESTART RECEIVE
REQID(*requestor ID*)

Example

Restart the receive of data from the network defined in mailbox (requestor) profile **DEPTA7F**.

```
PERFORM RESTART RECEIVE
    WHERE REQID(DEPTA7F)
```

RESTART SEND command

This command is used when sending EDI data, and is valid only when both the following are true:

- The network supports restart
- You specified checkpoint-level recovery when initially sending the data

If an error causes network processing to enter a restart situation while processing EDI data, this command can be used to restart and complete the send. For more information on checkpoint recovery, see the *GXS Expedite Base/MVS Programming Guide*.

Notes:

1. The envelope file and the requestor ID must be the same values specified on the initial send.
2. This command is not supported in the CICS environment.

Syntax

RESTART SEND
CLEARFILE(*clear specified file contents*)
ENVTYPE(*transaction envelope type*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SCRIPT(*script ID*)
SEQNUM(*increment network profile member numbers*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example

Restart the send of data from file **NETQUEUE** to the network defined in mailbox (requestor) profile **IINREQ**.

```
PERFORM RESTART SEND
    WHERE REQID(IINREQ) FILEID(NETQUEUE)
```

RETRANSLATE TO APPLICATION command

This command provides the same functions as the TRANSLATE TO APPLICATION command for transactions that were previously translated. It takes EDI documents from the Document Store, translates them to application format, and places the results in the file specified by the data format or in the override file specified by the trading partner usage/rule for translating the transaction. The data can be formatted as C and D records or as raw data.

Syntax

RETRANSLATE TO APPLICATION
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
ASSERTLVL(*session assertion level*)
BATCH(*translated transaction batch ID*)
BATCHSET(*set transaction batch ID*)
CCEXCEPTION(*job-step condition code*)
DIERRFILTER(*initial error filter set*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
EXTENDC(*translate with extended C record format*)
FORCETEST(*force test usage*)
FORMAT(*data format ID*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETID(*network ID*)
OPTRECS(*optional record type*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
SETCC(*condition codes*)
STDTRID(*EDI standard transaction set ID*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)

TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Re-translate all EDI documents in the Document Store that were received on December 14, 2001 from trading partner **PISCES**. Place the data in the application file specified by the data format or in the override file specified in the receive usage/rule for the trading partner.

```
PERFORM RETRANSLATE TO APPLICATION
    WHERE TRXDATE(01/12/14) TPNICKN(PISCES)
```

SAP STATUS EXTRACT command

This command extracts SAP status information from the Document Store.

Syntax

SAP STATUS EXTRACT
EXTRACTED(*previously extracted status*) Default is N.
OUTFILE(*output file name*)
OUTTYPE(*output file type*)
CLIENT(*SAP client ID*)
SAPSTAT(*SAP status starting value*) TO(*SAP status ending value*)
DAYS(*record starting date*) TO(*record ending date*)

Example

Extract SAP status records from the WebSphere Data Interchange database and write them to the file specified in the keywords `OUTFILE` and `OUTTYPE`.

```
PERFORM SAP STATUS EXTRACT
    WHERE OUTFILE() OUTTYPE() CLIENT() SAPSTAT() TO() DAYS() TO()
```

SAP STATUS REMOVE command

This command removes SAP status information from the Document Store.

Syntax

SAP STATUS REMOVE
CLIENT(*SAP client ID*)
EXTRACTED(*previously extracted status*) Default is Y.
PRIORTO(*Date before which to remove records*)
SAPSTAT(*SAP status starting value*) TO(*SAP status ending value*)
DAYS(*Record starting date*) TO(*Record ending date*)

Example

Remove SAP status information from the WebSphere Data Interchange database for all clients with status **16**.

```
PERFORM SAP STATUS REMOVE
WHERE SAPSTAT(16) TO(19)
```

SEND command

This command sends EDI transactions from an envelope file to the network. The envelope file is either the TD queue specified in the network profile member or a file specified in the command.

Syntax

SEND
CLEARFILE(*clear specified file contents*)
ENVTYPE(*transaction envelope type*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
ONEMSG(*read only one MQ message*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SCRIPT(*script ID*)
SEQNUM(*increment network profile member numbers*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Example 1

Send the transactions in the TD queue. Use the network specified by mailbox (requestor) profile member **IINREQ**.

```
PERFORM SEND
    WHERE REQID(IINREQ)
```

Example 2

Send the EDIFACT transactions in the file **NETQUEUE**. Use the network specified by mailbox (requestor) profile member **IINREQ**.

```
PERFORM SEND
    WHERE REQID(IINREQ) FILEID(NETQUEUE) ENVTYPE(E)
```

SENDFILE command

This command sends application data files that do not need an interchange header or that do not contain EDI data.

Syntax

SENDFILE
CLEARFILE(*clear specified file contents*)
FILEID(*processing file ddname*)
IFCC(*override condition codes*)
MSGUCLASS(*override message user class*)
ONEMSG(*read only one MQ message*)
RAWDATA(*translate to raw data format*)
REQID(*requestor ID*)
SCRIPT(*script ID*)
SEQNUM(*increment network profile member numbers*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)

Mapping issues with RAWDATA keyword on SENDFILE

The use of the **RAWDATA** keyword on the **SENDFILE** command will cause WebSphere Data Interchange to strip C and D records from an existing DF file generated in a previous step. When WebSphere Data Interchange generates a DF file in raw-data format the structure ID is placed in the first column of the data. This is not done with C and D record data. You must map the structure ID as the first element in the data if you wish to see the structure name. This can be done using the mapping tool in the WebSphere Data Interchange client to place the structure ID anywhere in the data.

Example 1

Send the data in the file named **FLATFILE** to trading partner **MYTP**.

```
PERFORM SENDFILE
    WHERE REQID(ME) FILEID(FLATFILE) TPNICKN(MYTP)
```

Example 2

Send the data in the file named **BULK** to trading partner **YOURTP** with a network message user class of **SPECIAL**.

```
PERFORM SENDFILE
    WHERE REQID(ME) FILEID(BULK) TPNICKN(YOURTP) MSGUCLASS(SPECIAL)
```

START CONTINUOUS RECEIVE command

This command is available only in the CICS environment. Like the CICS transaction EDIR, this command can be used to start a single continuous receive or start all eligible continuous receives. A continuous receive can be started if:

- A valid requestor ID is specified in the continuous receive profile member.
- The continuous receive profile member is marked ACTIVE.
- The network ID in the associated mailbox (requestor) profile member is IINCICS.

Syntax

START CONTINUOUS RECEIVE
IFCC(*override condition codes*)
MEMBER(*member profile name*)
SETCC(*condition codes*)

Example 1

Start all eligible continuous receives.

```
PERFORM START CONTINUOUS RECEIVE
```

Example 2

Start a continuous receive for profile member **CRMEM**.

```
PERFORM START CONTINUOUS RECEIVE
    WHERE MEMBER(CRMEM)
```

STOP CONTINUOUS RECEIVE command

This command is available only in the CICS environment. Like the CICS transaction EDIS, this command can be used to stop a single continuous receive or stop all eligible continuous receives. A continuous receive can be stopped if:

- A valid requestor ID is specified in the continuous receive profile member.
- The continuous receive profile member is marked ACTIVE.
- The network ID in the associated mailbox (requestor) profile member is IINCICS.

Syntax

STOP CONTINUOUS RECEIVE
IFCC(*override condition codes*)
MEMBER(*member profile name*)
SETCC(*condition codes*)

Example 1

Stop all active continuous receives.

```
PERFORM STOP CONTINUOUS RECEIVE
```

Example 2

Stop the continuous receive for profile member **CRMEM**.

```
PERFORM STOP CONTINUOUS RECEIVE
    WHERE MEMBER(CRMEM)
```

TRADING PARTNER CAPABILITY DATA EXTRACT command

This command gathers data about the transaction capabilities of trading partners including:

- Which transactions have maps, sorted by trading partner
- The total number of transactions translated against these maps
- The total number of transactions with unacceptable error levels translated against these maps

With a report writer or custom program, you can use this command to create reports such as:

- Pre-Migration Status
- Trading Partner Implementation Status
- Trading Partner Testing Status
- Cumulative Transaction Activity by Trading Partner
- Where Used Maps by Trading Partner
- Trading Partners by Maps Used

Note: If your customer tables are not in the same database as your runtime tables, all data fields cannot be included in this report.

Syntax

TRADING PARTNER CAPABILITY DATA EXTRACT
ADDRLN1(*trading partner address line 1*)
ADDRLN2(*trading partner address line 2*)
CMMTLN1(*trading partner comment line 1*)
CMMTLN2(*trading partner comment line 2*)
CMPYNM(trading partner company name)
DAYS(*starting date*) TO(*ending date*)
DIR(*processing direction*)
DYNSQL(*use dynamic SQL*)
IFCC(*override condition codes*)
MAPID(*map name*)
SETCC(*condition codes*)
STDDESC*(EDI standard description*)
STDID(*EDI standard ID*)
STDLV(*EDI standard release level*)
STDTRID(*EDI standard transaction set ID*)
STDVR(*EDI standard version*)
TESTMODE(*test transaction mode*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
USERPGM(*user program*)

Example 1

Company X is implementing EDI with all suppliers to their Atlanta assembly plant. They have set a target date of 12/31 to have 100% EDI on several transactions. Suppliers

start out in test mode for each new transaction and, after 100 successful test transactions, are eligible to move into production. The EDI steering committee meets monthly to review progress and needs a report that shows information for each supplier, such as:

- All transactions sent and received
- When the supplier went into test for each transaction type
- Total number of transactions sent or received in test mode
- When the supplier went into production for each transaction type
- Total number of transactions sent or received in production mode

```
PERFORM TRADING PARTNER CAPABILITY DATA EXTRACT
    WHERE CMMTLN1(Atlanta) CMMTLN2(Supplier)
```

Figure 11 shows an example of a trading partner capability report.

Trading Partner Capability Report Data

Company NameID	Location	Int ID	Trx	Std. ID	Dir	ID	Started Testing	Test Trx's	Started Productn	Product Trx's
Global Services	Tampa, FL	233119	850	X12V2R1	S	NONPRODX12V2R1PO	07/03/01	101	08/15/01	262
Only1 Co.	Lima, OH	399129	850	X12V2R3	S	PRODSUPX12V2R3PO	08/15/01	71	*NODATE*	0
Pulse Services	Armonk, NY	997223	850	X12V2R3	S	PRODSUPX12V2R3PO	*NODATE*	0	*NODATE*	0
Handlebars Etc.	Raleigh, NC	877519	850	X12V2R3	S	PRODSUPX12V2R3PO	05/07/01	115	07/02/01	554
			855	X12V2R3	R	PRODSUPX12V2R3PA	05/22/01	109	07/26/01	445
			856	X12V2R3	R	PRODSUPX12V2R3SN	06/13/01	187	07/22/01	1878
			861	X12V2R3	S	PRODSUPX12V2R3RA	08/08/01	242	*NODATE*	0
IBM	Dallas, TX	660599	850	X12V2R3	S	PRODSUPX12V2R3PO	01/08/01	126	03/14/01	927
			855	X12V2R3	R	PRODSUPX12V2R3PA	08/22/01	054	*NODATE*	0
			856	X12V2R3	R	PRODSUPX12V2R3SN	04/06/01	378	*NODATE*	0
Pen&Paper	Carson, CA	523567	850	X12V2R3	S	PRODSUPX12V2R3PO	02/02/01	113	03/02/01	856
			855	X12V2R3	R	PRODSUPX12V2R3PA	03/26/01	104	03/12/01	535
			856	X12V2R3	R	PRODSUPX12V2R3SN	03/13/01	127	11/23/01	278
			861	X12V2R3	S	PRODSUPX12V2R3RA	04/08/01	102	06/08/01	322

Figure 11. Example of a trading partner capability report

Example 2

You want to extract information associated with migrating the purchase order transaction from X12V2R1 to X12V3R1. Create a list of all the trading partners and usages or rules you must migrate to the new EDI standard.

```
PERFORM TRADING PARTNER CAPABILITY DATA EXTRACT
    WHERE STDID(X12V2R1) STDTRID(850)
```

TRADING PARTNER PROFILE DATA EXTRACT command

This command provides detailed information about your trading partners including:
- Company name
- Address
- Contacts
- Telephone number
- Nickname
- Network name
- Interchange ID
- Account ID
- User ID
- Latest data transmission date, sorted by trading partner

Before running this command, you must run the UPDATE STATISTICS command to gather the data.

With a report writer or custom program, you can use this command to create reports such as:
- Simple trading partner lists
- Comprehensive trading partner lists
- Trading partners sorted by division
- Inactive trading partners (no activity for *x* period of time)

Syntax

TRADING PARTNER PROFILE DATA EXTRACT
ACCTID(*network account ID*)
ADDRLN1(*trading partner address line 1*)
ADDRLN2(*trading partner address line 2*)
CMMTLN1(*trading partner comment line 1*)
CMMTLN2(*trading partner comment line 2*)
CMPYNM(trading partner company name)
CNTCTNM(*trading partner contact name*)
CNTCTPH(*trading partner contact phone number*)
DYNSQL(*use dynamic SQL*)
GRPCTLNO(*starting sender's group control nbr.*)
TO(*ending sender's group control nbr.*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTID(*trading partner interchange ID*)
LASTTRXDATE(*starting latest transaction date*)
TO(*ending latest transaction date*)
NETID(*network ID*)
SETCC(*condition codes*)
TPNICKN(*trading partner nickname*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
USERID(*network user ID*)
USERPGM(*user program*)

Example 1

Your EDI Administrator has requested that you provide a hard copy report every Monday morning that shows the trading partners who have sent data within the last six months to the Metal Finishing and Metal Shipping Divisions. This report is used to interface with the various departments and to contact the trading partner in case of trouble. The report contains:

- Company name and location
- Supplier/customer Dun and Bradstreet (DUNS) number
- EDI contact's name and phone number
- Date of the last transaction sent to them
- Control numbers of the last transaction sent to them

```
PERFORM TRADING PARTNER PROFILE DATA EXTRACT
    WHERE CMMTLN2(Metal Finishing) LASTTRXDATE(*-180) TO(*)
    WHERE CMMTLN2(Metal Shipping) LASTTRXDATE(*-180) TO(*)
```

Or, if these are the only two divisions beginning with the word Metal, you can use this command:

```
PERFORM TRADING PARTNER PROFILE DATA EXTRACT
    WHERE CMMTLN2(Metal*) LASTTRXDATE(*-180) TO(*)
```

Figure 12 shows an example of a trading partner profile report.

```
                    Trading Partner Information Report Data


Company Location   IntID   Net  Accnt  UserID     Contact      Contact       LastTrx   Ctrl No.
 Name                                             Name         Phone         Date
-------------------------------------------------------------------------------------------
A0 Corp Morris,MN  333219  NETA MMIDOO  55532214  Sam Heusen   813–555–5668  08/01/01  00000171
IBM     Tampa, FL  200762  NETA ADV001  99665593  Deb Garrie   762–555-2328  08/15/01  00990171
Alloy   Chicago,IL 543189  NETA ALUM00  18654771  Lou Green    619–555-7784  08/15/00  00044573
Inc.
DeKant  Queens, NY 337690  IIN  DECANT  USER01    Bob Kling    252–555-9006  11/06/01  00670170
Systems London, UK 856690  IIN  DECAHH  USER02    Joe Miller   413–555–5555  03/17/01  00006392
        London, UK 009210  IIN  DECA00  USER03    Heather Liu  413–555-2256  03/27/01  00066392

IBM     Paris   FR 665767  IIN  IBMIM   01NYC     Mary Lockler 919–555-8585  02/22/01  00601054
        Raleigh,NC 976554  IIN  IB      SANJOSE   Jane Hart    878–555-2876  02/06/01  00070378

MMM     Edison, NJ 332323  NETA MMM01   MMADMIN   Jon Murphy   828–555-9058  03/18/01  05567836
Motors  Miami,  FL 003292  IIN  MMM02   MMTECH    Pauline Gold 813–555-8421  01/22/01  00650054
```

Figure 12. Example of a trading partner profile report

Example 2

Generate a data extract of all the trading partners for your **ABC** and **DEF** divisions with whom you have not exchanged any transactions within the last six months. Print only information about trading partners on the network. Next, put the division in the `Comment Line 2` field of the trading partner profile.

```
PERFORM TRADING PARTNER PROFILE DATA EXTRACT
    WHERE CMMTLN2(ABC) LASTTRXDATE(*-999) TO(*-180) NETID(IN*)
    WHERE CMMTLN2(DEF) LASTTRXDATE(*-999) TO(*-180) NETID(IN*)
```

TRANSACTION ACTIVITY DATA EXTRACT command

This command collects data about the daily transaction activity of trading partners including:

- Total number of transactions sent or received on any day or range of days, sorted by ID
- Total number of transactions with errors that were sent or received on any day or range of days, sorted by ID

You can use this command to create reports such as:

- Outbound Document Audit Summary (overview)
- Inbound Document Audit Summary (detailed
- Transaction Activity by Transaction
- Transaction Activity by Trading Partner
- Outbound/Inbound Errors
- Activity Summary by Trading Partner or by Transaction

Syntax

TRANSACTION ACTIVITY DATA EXTRACT
ADDRLN1(*trading partner address line 1*)
ADDRLN2(*trading partner address line 2*)
CMMTLN1(*trading partner comment line 1*)
CMMTLN2(*trading partner comment line 2*)
CMPYNM(trading partner company name)
DAYS(*starting date*) TO(*ending date*)
DIR(*processing direction*)
DYNSQL(*use dynamic SQL*)
IFCC(*override condition codes*)
MAPID(*map name*)
SETCC(*condition codes*)
STDDESC*(EDI standard description*)
STDID(*EDI standard ID*)
STDLV(*EDI standard release level*)
STDTRID(*EDI standard transaction set ID*)
STDVR(*EDI standard version*)
TESTMODE(*test transaction mode*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
USERPGM(*user program*)

Example

Extract a list of all your trading partners and the number of invoices they sent you through EDI in the first six months of 2001. This report can be used to spot-check the accounts receivable system to verify that the invoice receipt process is in control. You receive both EDIFACT and X12 invoices. The transaction activity report provides the necessary data listed by trading partner.

```
PERFORM TRANSACTION ACTIVITY DATA EXTRACT
    WHERE STDTRID(INVOIC) DAYS(01/01/01) TO(01/06/30) DIR(R)
    WHERE STDTRID(810)    DAYS(01/01/01) TO(01/06/30) DIR(R)
```

Figure 13 shows an example of a transaction activity data extract report, after the data has been sorted and formatted.

```
              EDI Invoice Activity Summary for 1H01

Compamy                                                    Trx  Errors  Trx  Errors
Name        Location    Int ID   Trx ID   Std. ID   Dir   MTD   MTD    YTD  YTD
----------  ----------  ------   ------   --------  ---   ---   ---    ---  ------
IBM Global  Tampa, FL   933489   810      X12V2R1   R      14    1     136    9
Services

CULATER     Dallas, TX  322217   810      X12V2R3   R      49    5     450    8
                                 810      X12V2R3   R      10    5      90   14
                                 810      X12V2R3   R     250   23    2200   28
                                 INVOIC   EDIL921   R      24    0     224    0

Definite    Paris, FR   005901   810      X12V2R3   R     456    0    4104    0
Inc.                             810      X12V2R3   R      45    0     405    0
                                 INVOIC   X12V2R3   R     789    0    7101    0

Jay's       Perry, NC   555489   810      X12V2R3   R    102     2     918    7
Turbo                            810      X12V2R3   R      83    3     747   12
                                 810      X12V2R3   R      89    0     789    0
                                 810      X12V2R3   R     105    0     945    0

Cool Pool   Carmel, CA  748723   810      X12V2R3   R     234    0    1245    2
Service                          810      X12V2R3   R     216    0    1245    4
```

Figure 13. Example of a formatted transaction activity report

TRANSACTION DATA EXTRACT command

This command extracts detailed information about transactions, sorted by transaction handle. You can use this command to create report data to:

- Report on the number of purchase orders sent in a given period of time
- Report on the total number of bytes sent in a given period of time (this can be useful for charging back costs to other departments based on their EDI usage)
- Create a customized functional acknowledgment tracking report by application or by department; for example, an exception report on purchase orders sent more than 2 days ago that have not been acknowledged
- Create an exception report flagging missing control numbers for inbound envelopes

You can also use this command for functions other than reporting, such as:

- Archiving Document Store data
- Loading status data directly into the application by application key; for example, the status for each invoice sent can be loaded into the billing system by invoice number

To extract detailed information about enveloped transactions, use the ENVELOPE DATA EXTRACT command (see 37).

Syntax

TRANSACTION DATA EXTRACT
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLICATION(*write application data record*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
CONCATENATE(*concatenate extract data*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GROUP(*write group data record*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
IMAGE(*write image data record*)

INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTERCHANGE(*write interchange data record*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
LOGFAMSG(*user defined code*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
RECEIVEACKDATA(*write detailed acknowledgment data*)
RECEIVEACKIMAGE(*write receive acknowledgment record*)
SENDACKDATA(*write detailed acknowledgment data*)
SENDACKIMAGE(*write acknowledgment record*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRANSACTION(*write transaction data record*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
USERPGM(*user program*)

Example 1

Retrieve the interchange, group, and transaction information for all transactions with trading partner name **PISCES** and interchange control number **000008888**. Place the results in the EDIQUERY file.

```
PERFORM TRANSACTION DATA EXTRACT
    SELECTING INTERCHANGE(Y) GROUP(Y) TRANSACTION(Y)
    WHERE TPNICKN(PISCES) INTCTLNO(000008888) DIR(S)
```

Example 2

Retrieve the interchange, group, and transaction information for all transactions with application sender ID **SERVO**. Include application and image information.

```
PERFORM TRANSACTION DATA EXTRACT
    SELECTING INTERCHANGE(Y) GROUP(Y) TRANSACTION(Y) APPLICATION(Y) IMAGE(Y)
    WHERE APPSNDID(SERVO) DIR(S)
```

TRANSFORM command

You can use this command to translate data in any format to any other format defined in your WebSphere Data Interchange system. This command uses data transformation maps to translate the data.

Syntax

TRANSFORM
ACFIELD(*starting application control field data*)
    TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
BATCHSET(*set transaction batch ID*)
CLEARFILE(*clear specified file contents*)
DEENVONLY(*deenvelope only indicator*)
DICTIONARY(*input dictionary name*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*)
    TO(*ending delivery date*)
DLVTIME(*starting delivery time*)
    TO(*ending delivery time*)
DOCUMENT(*input data document name*)
DUPENV(*process duplicate envelopes*)
ENCODETARGET(*supported encoding value*)
ENVDATE(*starting transaction envelope date*)
    TO(*ending transaction envelope date*)
ENVELOPE(*envelope indicator*)
ENVTIME(*starting transaction envelope time*)
    TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
    TO(*ending transaction purge date*)
FADELAY(*delay queuing functional acknowledgment*)
FILTERMSGS(*filter error messages*)
FORMAT(*data format ID*)
FUNACKFILE(*functional acknowledgment file name*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLLEN(*group control number length*)
GRPCTLNO(*starting sender's group control number*)
    TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*)
IFCC(*override condition codes*)
IGNOREBOM (*ignore byte order mark*)
IGNOREINFO(*ignore informational error messages*)
IGNOREWARN(*ignore warning error messages*)
INFILE(*input data file name*)
INTCTLLEN(*interchange control number length*)
INTCTLNO(*starting sender's interchange control number*)

TO(*ending sender's interchange control number*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
INTYPE(*input data file type*)
MAPID(*map name*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
ONEMSG(*read only one MQ message*)
OPTRECS(*optional record type*)
OUTFILE(*output data file name*)
OUTLEN(*maximum output record length*)
OUTTYPE(*output data file type*)
PAGE(*page indicator*)
PAGETHRESHOLD(*page threshold*)
PARSEFILE(*parse from file indicator*)
PURGINT(*purge interval*)
RECOVERY(*recovery unit of work*)
SAPFILE(*SAP file name*)
SAPTYPE(*SAP file type*)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*)
TO(*ending request sent date*)
SNDTIME(*starting request sent time*)
TO(*ending request sent time*)
SOURCEENCODE(*supported encoding value*)
STDTRID(*EDI standard transaction set ID*)
SYNTAX(*input data syntax type*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRACELEVEL(*trace level*)
TRXCTLLEN(*transaction control number length*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*)
TO(*ending transaction date*)
TRXSTAT(*starting transaction status*)
TO(*ending transaction status*)
TRXTIME(*starting transaction time*)
TO(*ending transaction time*)
XMLDTDS(*XML DTD path*)
XMLEBCDIC(*EBDCDIC indicator*)
XMLNS(*XML namespace indicator*)
XMLSCHEMAVAL(*XML schema validation*)
XMLSPLIT(*XML Split document*)
XMLVALIDATE(*XML validation indicator*)
XRECS(*Generate X records for C and D*)

Example

Transform the MQSeries queue file **PURCH1** and output the results in file **PURCHTRN**.

```
PERFORM TRANSFORM
WHERE INFILE(PURCH1) INTYPE(MQ) SYNTAX(X) OUTFILE(PURCHTRN)
```

TRANSLATE AND ENVELOPE command

This command combines the functions of the TRANSLATE TO STANDARD and ENVELOPE commands. This command takes EDI transactions from an application file, translates them to EDI format, places the results in the Document Store, and creates envelopes for the transactions. Then, the enveloped transactions are placed in the TD queue or in a file specified in the command. Processing is quicker than using the separate commands, but optimum enveloping is best achieved with the separate commands.

Syntax

TRANSLATE AND ENVELOPE
APPFILE(*application data file name*)
APPSEC(*application file security level*)
APPTYPE(*application file type*)
BATCHSET(*set transaction batch ID*)
DELFILE(*delete application file associated with logical filename)*)
DIERRFILTER(*initial error filter set*)
ENVPRBREAK(*start new envelope*)
EXTENDC(*translate with extended C record format*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
IACCESS(*IEXIT access*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
ITPBREAK(*new interchange envelope*)
ITYPE(*IEXIT program type*)
ONELOGICAPP(*single logical file*)
ONEMSG(*read only one MQ message*)
OPTRECS(*optional record type*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RAWDATA(*translate to raw data format*)
RAWFMTID(*raw data format ID*)
RAWUSAGE(*raw data transaction type*)
RECOVERY(*recovery unit of work*)
SAPUPDT(*track SAP status*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
TPID(*trading partner ID*)
XML(*XML required*)
XMLDICT(*XML dictionary address*)
XMLDTDS(*XML DTD path*)

Example 1

Translate and envelope the transactions in ddname **APPDATA**. Place the enveloped transactions in the TD queue specified by the network profile. Return the information record created during translation.

```
PERFORM TRANSLATE AND ENVELOPE
    WHERE APPFILE(APPDATA) OPTRECS(I)
```

Example 2

In CICS, translate and envelope the transactions in TD queue **AP01**. Place the enveloped transactions in TS queue named **ENVDATA**. Return the information record and envelope headers created during translation.

```
PERFORM TRANSLATE AND ENVELOPE
    WHERE APPFILE(AP01) APPTYPE(TD) FILEID(ENVDATA) OPTRECS(IE)
```

TRANSLATE AND SEND command

This command combines the functions of the TRANSLATE TO STANDARD, ENVELOPE, and SEND commands into one step. Processing is quicker than using the separate commands, but optimum enveloping is best achieved by using the separate commands.

Note: This command will not work properly with direct connection networks such as point-to-point if the application data file contains transactions for more than one trading partner.

Syntax

TRANSLATE AND SEND
APPFILE(*application data file name*)
APPSEC(*application file security level*)
APPTYPE(*application file type*)
ASSERTLVL(*session assertion level*)
BATCHSET(*set transaction batch ID*)
CLEARFILE(*clear specified file contents*)
DELFILE(*delete application file associated with logical filename)*)
DIERRFILTER(*initial error filter set*)
ENVPRBREAK(*start new envelope*)
EXTENDC(*translate with extended C record format*)
FILEID(*processing file ddname*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
IACCESS(*IEXIT access*)
IAREA(*IEXIT information*)
IEXIT(*interchange control program*)
IFCC(*override condition codes*)
INMEMTRANS(*transactions in memory*)
ITPBREAK(*new interchange envelope*)
ITYPE(*IEXIT program type*)
MSGUCLASS(*override message user class*)
ONELOGICAPP(*single logical file*)
ONEMSG(*read only one MQ message*)
OPTRECS(*optional record type*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RAWFMTID(*raw data format ID*)
RAWTEST(*raw test data*)
RAWUSAGE(*raw data transaction type*)
RECOVERY(*recovery unit of work*)
REQID(*requestor ID*)
SAPUPDT(*track SAP status*)
SCRIPT(*script ID*)
SEQNUM(*increment network profile member numbers*)
SERVICESEGVAL(*service segment validation level*)
SETCC(*condition codes*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)

Example 1

Translate and send the transactions in file **APPDATA**. Assign batch number **121401** to these transactions.

```
PERFORM TRANSLATE AND SEND
    WHERE APPFILE(APPDATA) REQID(IINREQ) BATCHSET(121401)
```

Example 2

Translate and send the transactions in file **RAWFILE**. Data format **POSEND** describes these transactions, which are in raw data format.

```
PERFORM TRANSLATE AND SEND
    WHERE APPFILE(RAWFILE) REQID(IINREQ) RAWFMTID(POSEND)
```

TRANSLATE TO APPLICATION command

This command takes EDI documents from the Document Store, translates them to application format, and places the results in the file specified by the data format or in the override file specified by the trading partner usage/rule for translating the transaction. The data can be formatted as C and D records or as raw data.

Note: In CICS, you can also deliver the data to a program or CICS transaction, as specified in the `Application file name` and `Application file type` fields of the data format.

Syntax

TRANSLATE TO APPLICATION
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
ASSERTLVL(*session assertion level*)
BATCHSET(*set transaction batch ID*)
CCEXCEPTION(*job-step condition code*)
DIERRFILTER(*initial error filter set*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
EXTENDC(*translate with extended C record format*)
FORCETEST(*force test usage*)
FORMAT(*data format ID*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
MULTIDOCS(*multiple-document file*)
NETID(*network ID*)
OPTRECS(*optional record type*)
PAGE(*pageable translation*)
RAWDATA(*translate to raw data format*)
SETCC(*condition codes*)
STDTRID(*EDI standard transaction set ID*)
TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)
XML(*XML required*)

XMLDICT(*XML dictionary address*)
XMLDTDS(*XML DTD path*)
XMLEBCDIC(*EBDCDIC indicator*)
XMLSEGINP(*line break indicator*)
XMLSTDID(*destination EDI standard ID*)
XMLVALIDATE(*XML validation indicator*)

Example 1

Translate all EDI documents in the Document Store that were received on December 14, 2001 with network ID **IINR41**. Place the data in the application file specified by the data format or in the override file specified in the receive usage/rule for the trading partner.

```
PERFORM TRANSLATE TO APPLICATION
    WHERE TRXDATE(01/12/14) NETID(IINR41)
```

Example 2

Translate all EDI documents in the Document Store that were received between December 14, 2001 and today's date. Place the data in the application file specified by the data format or in the override file specified in the receive usage/rule for the trading partner.

```
PERFORM TRANSLATE TO APPLICATION
    WHERE TRXDATE(01/12/14) TO(*)
```

Example 3

Translate all EDI documents in the Document Store that were received on December 14, 2001 with network ID **IINR41**. Also translate all EDI data received on December 14, 2001 from trading partner **PISCES**. Place the data in the application file specified by the data format or in the override file specified in the receive usage/rule for the trading partner.

```
PERFORM TRANSLATE TO APPLICATION
    WHERE TRXDATE(01/12/14) NETID(IINR41)
    WHERE TRXDATE(01/12/14) TPNICKN(PISCES)
```

TRANSLATE TO STANDARD command

This command translates application data to an EDI standard format and places the results in the Document Store. The application data can be in C and D record format or in raw data format.

Syntax

TRANSLATE TO STANDARD
APPFILE(*application data file name*)
APPSEC(*application file security level*)
APPTYPE(*application file type*)
ASSERTLVL(*session assertion level*)
BATCHSET(*set transaction batch ID*)
DELFILE(*delete application file associated with logical filename)*)
DIERRFILTER(*initial error filter set*)
EENVDATE(*earliest transaction enveloping date*)
EXTENDC(*translate with extended C record format*)
FIXEDFILEID(*fixed-to-fixed output ddname*)
IFCC(*override condition codes*)
ONEMSG(*read only one MQ message*)
OPTRECS(*optional record type*)
PAGE(*pageable translation*)
PURGINT(*purge interval*)
RAWDATA(*translate to raw data format*)
RAWFMTID(*raw data format ID*)
RAWUSAGE(*raw data transaction type*)
SAPUPDT(*track SAP status*)
SETCC(*condition codes*)
TPID(*trading partner ID*)
XML(*XML required*)
XMLDICT(*XML dictionary address*)
XMLDTDS(*XML DTD path*)

Example 1

Translate the transactions in file **INVOICES** from a raw data format to an EDI standard format and place the results in the Document Store. The data format **SENDINVOICES** describes the raw data.

```
PERFORM TRANSLATE TO STANDARD
    WHERE APPFILE(INVOICES) RAWFMTID(SENDINVOICES)
```

Example 2

Translate the transactions in two application files from C and D record format to EDI standard format and place the results in the Document Store. Assign batch ID **121401** to these transactions.

```
PERFORM TRANSLATE TO STANDARD
    WHERE APPFILE(APDATA01) BATCHSET(121401)
    WHERE APPFILE(APDATA02) BATCHSET(121401)
```

UNLOAD LOG ENTRIES command

This command reads all the entries in the event log associated with the `Application ID`. The entries selected for removal are copied to the archive file, and all other entries associated with the `Application ID` are copied to the hold file.

Event log entries that are associated with active or held transactions in the Document Store are not eligible for archive and are not selected for removal. The entries selected for removal are copied to the archive file and immediately deleted.

To ensure concurrency, set the number of deletes (`Numdels`) performed before a COMMIT is issued to a relatively low value. This command can be run as the first step of a multi-step process. (See the table on 7.)

Syntax

UNLOAD LOG ENTRIES
APPLID(*application ID*)
ARCHIVEFILE(*event log archive file name*)
ARCHIVETYPE(*event log archive file type*)
HOLDFILE(*event log hold file name*)
HOLDTYPE(*event log hold file type*)
IFCC(*override condition codes*)
LOGAEID(*starting event log associated entry ID*) TO(*ending event log associated entry ID*)
LOGDATE(*starting event log date*) TO(*ending event log date*)
LOGFORM(*starting event log format ID*) TO(*ending event log format ID*)
LOGTIME(*starting event log time*) TO(*ending event log time*)
LOGUSER(*starting event log user ID*) TO(*ending event log user ID*)
NUMDELS(*number of database deletes before commit*)
SETCC(*condition codes*)

Example

Unload log entries from the application log file for application EDIFFS dated December 14, 2001.

```
PERFORM UNLOAD LOG ENTRIES
    WHERE APPLID(EDIFFS) LOGDATE(12/14/01)ARCHIVEFILE(ARCHTRAN)
      HOLDFILE (HOLDTRAN)
```

UNPURGE command

This command restores a transaction to the status it had before it was marked for purging unless the storage time for the transaction has expired. You can restore transactions marked for purging by the PURGE command until you execute the REMOVE TRANSACTIONS command. Once the REMOVE TRANSACTIONS command runs, you cannot restore the transactions. If a transaction is one of a related group, all transactions in the group are restored to their former status. If a storage time for the transaction has expired, it can still be eligible for purging. Running the UNPURGE command does not change the purge status of transactions whose storage time has expired.

The HOLD, PURGE, RELEASE, and UNPURGE commands share a common syntax.

Syntax

UNPURGE
ACFIELD(*starting application control field data*)
TO(*ending application control field data*)
APPLID(*application ID*)
APPRECID(*application receiver department ID*)
APPSNDID(*sender's department ID*)
BATCH(*translated transaction batch ID*)
DIR(*processing direction*)
DLVDATE(*starting delivery date*) TO(*ending delivery date*)
DLVTIME(*starting delivery time*) TO(*ending delivery time*)
ENVDATE(*starting transaction envelope date*)
TO(*ending transaction envelope date*)
ENVTIME(*starting transaction envelope time*)
TO(*ending transaction envelope time*)
ENVTYPE(*transaction envelope type*)
EPURDATE(*starting transaction purge date*)
TO(*ending transaction purge date*)
FORMAT(*data format ID*)
FUNACKP(*pending functional acknowledgment*)
GRPCTLNO(*starting sender's group control number*)
TO(*ending sender's group control number*)
HANDLE(*starting transaction ID*) TO(*ending transaction ID*)
IFCC(*override condition codes*)
INTCTLNO(*starting sender's interchange control nbr*)
TO(*ending sender's interchange control nbr*)
INTRECID(*interchange receiver ID*)
INTSNDID(*interchange sender ID*)
NETACKP(*pending network acknowledgment*)
NETID(*network ID*)
NETSTAT(*network transaction status*)
SETCC(*condition codes*)
SNDDATE(*starting request sent date*) TO(*ending request sent date*)
SNDTIME(*starting request sent time*) TO(*ending request sent time*)
STDTRID(*EDI standard transaction set ID*)
STSTAT(*transaction status*)

TPID(*trading partner ID*)
TPNICKN(*trading partner nickname*)
TRERLVL(*maximum translation error level*)
TRXCTLNO(*starting transaction set control number*)
TO(*ending transaction set control number*)
TRXDATE(*starting transaction date*) TO(*ending transaction date*)
TRXSTAT(*transaction processing status*)
TRXTIME(*starting transaction time*) TO(*ending transaction time*)

Example

Restore all EDI documents with application control numbers **PO112233** through **PO112244** that are marked for purging by user request.

```
PERFORM UNPURGE
    WHERE ACFIELD(PO112233) TO(PO112244) STSTAT(4)
```

UPDATE STATISTICS command

The UPDATE STATISTICS command is used to update the management reporting statistics from the pending statistics table to the reporting tables. You can use the NUMUPDTS keyword to limit the number of updates performed at one time.

Note: You must run this command before running management reports.

Syntax

UPDATE STATISTICS
NUMUPDTS(*number of database updates before commit*)

Example

Update the statistics tables before you run a management reporting data extract. Suppose you use the DB2 version of WebSphere Data Interchange . Because of application requirements, the system is running multiple simultaneous translation jobs and it is heavily loaded. Also, the database administrator has set the DB2 timeout value rather low. As a result, you are encountering DB2 timeouts. To reduce the amount of time that the UPDATE STATISTICS process holds table locks, set the NUMUPDTS value to a low number such as 25. This causes the UPDATE STATISTICS job to commit work and release all locks after every 25 DB2 updates.

```
PERFORM UPDATE STATISTICS
    WHERE NUMUPDTS(25)
```

UPDATE STATUS command

This command retrieves network acknowledgments for all mailbox (requestor) profile members, specified members, or specified network profile members. It processes all network acknowledgments that resulted from previous SEND commands, pairs the acknowledgments with the transactions originally sent, and updates transaction status.

The WHERE clause is optional for this command. If you do not specify a WHERE clause, all network profile members are processed.

You can request network acknowledgments in the `Net acknowledgment` field of the trading partner or mailbox (requestor) profile.

Syntax

UPDATE STATUS
IFCC(*override condition codes*)
NETID(*network ID*)
REQID(*requestor ID*)
SETCC(*condition codes*)

Example 1

Receive and process all network acknowledgments for all mailbox (requestor) profile members.

```
PERFORM UPDATE STATUS
```

Example 2

Receive and process network acknowledgments for all mailbox (requestor) profile members with network profile **IINB41**.

```
PERFORM UPDATE STATUS
    WHERE NETID(IINB41)
```

Example 3

Receive and process all network acknowledgments for mailbox (requestor) profile members, **ROBOX** and **KANBOX**.

```
PERFORM UPDATE STATUS
    WHERE REQID(ROBOX)
    WHERE REQID(KANBOX)
```

Chapter 3. Utility keywords

This chapter describes the keywords used with the WebSphere Data Interchange Utility commands.

ACCTID

The Network Account ID of the trading partner as entered in the `Account Number` field of the trading partner profile. The maximum length is 32.

This keyword is used with the following commands:

NETWORK ACTIVITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT

ACFIELD

The application control field or fields. Consists of either the `AC` field in the data format or the concatenation of data in the fields specified during transaction translation. For more information about the application control field, see the *WebSphere Data Interchange for MultiPlatforms User's Guide*, SC34-6215-01. Concatenated field data is used if any application control field names are specified in transaction translation. This field contains control numbers such as purchase order numbers and is case sensitive. The maximum length is 35.

Note: ACFIELD will be interpreted as ACFIELD1. See "ACFIELD*n*" on page 130 for more information.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS

RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATION TO APPLICATION
UNPURGE

ACFIELD*n*

The application control fields where *x* is a number 1 through 5.

Consists of either the AC field in the data format or the concatenation of data in the fields specified during transaction translation. For more information about the application control field, see the *WebSphere Data Interchange for MultiPlatforms User's Guide*, SC34-6215-01. Concatenated field data is used if any application control field names are specified in transaction translation. This field contains control numbers such as purchase order numbers and is case sensitive. The maximum length is 256.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATION TO APPLICATION
UNPURGE

ACKFILE

The ddname of a file containing network acknowledgments. If you do not specify this keyword, the value of the `Net output file` field in the associated network profile member is used.

This keyword is used only with the PROCESS NETWORK ACKS command.

Note: In CICS, WebSphere Data Interchange does not change the `ACKFILE` value to uppercase.

ACKTYPE

The acknowledgments file type used only for CICS and MQ (which is supported in both z/OS and CICS). Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name

TD Transient data queue

TM Temporary storage queue - main storage

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

VS VSAM entry sequenced data set

This keyword is used only with the PROCESS NETWORK ACKS command.

If you do not specify this keyword for z/OS, this field is ignored and the corresponding file name (the ddname of a sequential file) in the network profile member is used.

ACTUSAGE

Indicates how WebSphere Data Interchange should handle the activation of imported usages and rules. Valid values are:

NOREPL

An active imported usage/rule replaces an existing active usage/rule in the database. This is the default. If an active usage/rule that matches the key for the imported usage/rule already exists, the imported usage/rule is made inactive.

REPL An active imported usage/rule replaces an existing active usage/rule in the database. In this case, the existing usage/rule is made inactive and the imported usage/rule becomes the active usage.

FORCE

All imported usages/rules are forced to be active regardless of their current status, or the existence of another active usage/rule in the database. If an active usage/rule that matches the key for the imported usage/rule exists in the database, it is made inactive and the imported usage/rule becomes the active.

This keyword is used only with the IMPORT command.

ADDRLN1

The first line of the address as entered in the `Address line 1` field of the trading partner profile. This field is case sensitive. The maximum length is 40.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

ADDRLN2

The second line of the address as entered in the `Address line 2` field of the trading partner profile. This field is case sensitive. The maximum length is 40.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

AGENCY

The responsible agency code specified on the send or receive map usage. The maximum length is 8. The default is blank.

This keyword is used with the DELETE USAGE command.

APPFILE

When used with EDI processing commands, this keyword specifies the ddname of the file containing the application data. The data can be C and D records or raw data records. For more information on these record, see the . The maximum length is eight.

This keyword is used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND SEND
RECEIVE AND TRANSLATE
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

Note: In CICS, WebSphere Data Interchange does not change the `APPFILE` value to uppercase.

APPLICATION

Indicates whether an application data record is written to the EDIQUERY file. Valid values are:

Y Writes application data records

N Discards application data records. This is default.

Received transactions can be translated multiple times and will have an application record for each attempt. Send transactions only have one application record. For the format of these records, “Application data extract record layout” on page 319.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

APPLID

The application ID. When used as a selection criteria, this field specifies the application ID that was used to create the records being examined. (These records can be transactions in the Document Store or entries in the event log.) Valid values are:

EDIFFS
WebSphere Data Interchange Utility

EDIMP WebSphere Data Interchange Facility

(user-defined)
Your application APPLID

You can switch to another application ID by using the function codes for the WebSphere Data Interchange API. The maximum length is eight.

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
LOAD LOG ENTRIES
NETWORK ACTIVITY DATA EXTRACT
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2

PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE LOG ENTRIES
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
UNLOAD LOG ENTRIES
UNPURGE

APPLTPID

The application trading partner nickname specified on the send or receive map usage. The maximum length is 16.

This keyword is used with the DELETE USAGE command.

APPRECID

The application receiver ID. Identifies the specific department or business area in the receiver's company. The translator uses this ID first to attempt to locate the trading partner usage/rule for receive transactions. The maximum length is 35.

This keyword is used with the following commands:

DELETE USAGE
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION

TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

APPSEC

The security level of the application file. Indicates whether the application files specified with the APPFILE keyword are read only or read and write. Valid values are:

U Opens the application files for both read and write. The files can be opened for extend, and then closed. This is the default. This process ensures that each file has been properly initialized and that WebSphere Data Interchange does not process data that is not valid.

R Opens the application files in read only mode. If this option is chosen, you must make sure the application files are properly initialized.

This keyword is used with the following commands:

TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

APPSNDID

A specific department or business area in the sender's company. The maximum length is 35.

This keyword is used with the following commands:

DELETE USAGE
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSLATE TO APPLICATIONS
TRANSFORM
UNPURGE

APPTYPE

Indicates the application file type. Used only for CICS and MQ (which is supported in both z/OS and CICS). You can use this keyword to override the `Receive application file name` specified in the data format or specified on the Receive Transaction Usage Override panel. Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name

PG Response program (inbound processing only)

TD Transient data queue

TM Temporary storage queue - main storage

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

TX Response transaction (inbound processing only)

This keyword is used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND TRANSLATE
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

If you do not specify this keyword for z/OS, this field is ignored and the corresponding file keyword (the ddname of a sequential file) is used.

ARCHIVEFILE

The event log archive file name. Event log entries selected for removal are written to this file. The maximum length is eight.

This keyword is used with the following commands:

REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

ARCHIVETYPE

Indicates the event log archive file type used only for CICS and MQ (which is supported in both z/OS and CICS). Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name

TD Transient data queue

TM Temporary storage queue - main storage

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

This keyword is used with the following commands:

REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

If you do not specify this keyword for z/OS, this field is ignored and the corresponding file keyword (the ddname of a sequential file) is used.

ASSERTLVL

Specifies the level of assertions that are active for this translation session. Assertions are established during the translation process using the &ASSERT*n* special literals. The *n* can be a value from **0** to **9**, which establishes 10 assertion levels. Only &ASSERT requests with an *n* value greater than or equal to the ASSERTLVL value will be evaluated. For example, when ASSERTLVL(**5**) is specified, only the literals &ASSERT5, &ASSERT6, &ASSERT7, &ASSERT8, and &ASSERT9 will be evaluated. Level 9 assertions (&ASSERT9) are always active.

This keyword is used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND TRANSLATE
RETRANSLATE TO APPLICATION
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

BATCH

Specifies the batch ID assigned to a transaction when it was translated. You can use this keyword to select only transactions with a particular batch ID. The maximum length is eight.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE

QUERY
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

BATCHSET

Specifies that a user-defined ID be assigned to the transaction when it is translated. You can use this keyword to identify transactions you want to retrieve as a group. The transactions remain independent. An action performed on one transaction does not affect other transactions with the same batch ID. The maximum length is eight. The default batch ID is a date and time stamp in the format `ddhhmmss`.

This keyword is used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND TRANSLATE
RETRANSLATE TO APPLICATION
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

CCEXCEPTION

Indicates whether job-step condition code 0003 or 0000 is returned when application data is written to the exception file. Valid values are:

Y Generates a minimum job step condition code of 0003 if application data is written to the exception file.

N Generates a job-step condition code of 0000 if translation completes successfully and the application data was written to the exception file. This is the default. Does not report application data written to the exception file as an error.

X Generates a job-step condition code of 903 to distinguish this from other errors that result in a 0003 condition code.

This keyword is used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND TRANSLATE
RETRANSLATE TO APPLICATION
TRANSLATE TO APPLICATION

CLEARFILE

Indicates whether the specified file is cleared before it is used for further processing.

For the TRANSFORM command, this keyword indicates whether the output file or queue is cleared before writing the translated output. Valid values are:

Y Clears the file before writing the translated output

N Does not clear the file before writing the translated output. This is the default.

For the all other commands, this keyword indicates whether the receive file is cleared after a send is completed or before a receive is issued. Valid values are:

Y Clears the file after a send is completed or before a receive is issued

N Does not clear the file after a send is completed or before a receive is issued. This is the default.

This keyword is used with the following commands:

ENVELOPE AND SEND
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND SEND
RECEIVE AND TRANSLATE
RECVFILE AND SEND
REENVELOPE AND SEND
SEND
TRANSFORM
TRANSLATE AND SEND

Note: This keyword is ignored for batch receive processes. The file allocation disposition in the batch JCL will determine if the TD queue file is cleared.

CLIENT

Indicates which SAP status records are extracted or removed using the SAP client ID. The default is **ALL**.

This keyword is used with the following commands:

SAP STATUS EXTRACT
SAP STATUS REMOVE

CMMTLN1

The first comment line as entered in the Comment line 1 field of the trading partner profile. This field is case sensitive. The maximum length is 40.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

CMMTLN2

The second comment line as entered in the Comment line 2 field of the trading partner profile. This field is case sensitive. The maximum length is 40.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

CMPYNM

The company name as entered in the Company name field of the trading partner profile. This field is case sensitive. The maximum length is 40.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

CNTCTNM

The contact name as entered in the Contact name field of the trading partner profile. This field is case sensitive. The maximum length is 30.

This keyword is used only with the TRADING PARTNER PROFILE DATA EXTRACT command.

CNTCTPH

The contact phone number as entered in the Contact phone number field of the trading partner profile. The maximum length is 25.

This keyword is used only with the TRADING PARTNER PROFILE DATA EXTRACT command.

CONCATENATE

Indicates whether requested data extract information is written as separate records, or concatenated and written as a single record. Valid values are:

Y Concatenates categories and writes them as a single record

N Writes information as separate records. This is the default.

If concatenation is requested, the following hierarchy is used:
1. Interchange
2. Group
3. Transaction
4. Application

Each application entry written for a transaction is included with duplicate interchange, group, and transaction information.

Note: Concatenation does not apply to image records. Transaction and acknowledgment images are always written as separate records. Detailed acknowledgment data is always concatenated.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

CTLFILE

The ddname of an import or export control file that describes what data is to be imported or exported. For information about the control file format, see the table on 214.

This keyword is used with the following commands:

EXPORT
IMPORT

CTLTYPE

Indicates the import or export control file type. Used only for CICS and MQ (which is supported in both z/OS and CICS). Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name

TD Transient data queue

TM Temporary storage queue - main storage

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

This keyword is used with the following commands:

EXPORT
IMPORT

If you do not specify this keyword for z/OS, this field is ignored and the default for the corresponding file keyword is the ddname of a sequential file.

DAYS

A single date, or the start date of a date range. If the start date of a date range, this keyword must be followed by the `TO` keyword with an ending value.

NETWORK ACTIVITY DATA EXTRACT
SAP STATUS EXTRACT
TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

DEENVONLY

If set to Y, this means the data will only be deenveloped and put in document store, and will not be translated. This default is N. (Functionally equivalent to PERFORM DEENVELOPE.)

This keyword is used with the TRANSFORM command.

DELFILE

If **DELFILE** is specified on a **PERFORM TRANSLATE TO STANDARD**, **PERFORM TRANSLATE AND ENVELOPE** or a **PERFORM TRANSLATE AND SEND** command, the application file associated with the logical filename will be deleted once a successful translation is complete.

If an error occurs and the DF cannot be translated then the file is not deleted.

Note: **DELFILE** will delete the file on AIX® and Windows® but on z/OS the file will be emptied of its contents rather than being deleted.

DICTIONARY

Specifies the dictionary name for the input data. For application data, this field is required. For EDI and XML data, this field will override values extracted from the data. For EDI data, if you do not specify this keyword, the values in GS08 (X12) and UNH02 (EDIFACT) will be used to determine the value for this field in the EDI2DICT translation table shipped with WebSphere Data Interchange . For XML data, if you do not specify this keyword, the root element will be used to determine the dictionary name.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
TRANSFORM

DIERRFILTER

Specifies the initial set of errors to filter for this translation session. To override the filters defined in a usage/rule, you can use the DIERRFILTER keyword with a value of **IGNORE** to tell WebSphere Data Interchange to ignore all reported errors. This enables you to view all errors for a particular session without changing the map generally used. For more information on error filtering, see “Error filtering” on page 21 or the field definition for ERRFILTER in the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
RETRANSLATE TO APPLICATION
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

DIR

Indicates the direction of the transmission. Valid values are:

R Receiving

S Sending

This keyword is used with the following commands:

DELETE USAGE (default R)
DOCUMENT DATA EXTRACT
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE

PURGE
QUERY
RECONSTRUCT
RECONSTRUCT AND SEND
RELEASE
REMOVE TRANSACTIONS
TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

DLVDATE

Specifies the date, or a range of dates, when the transactions you want to work with were delivered to the application. If this is the start date of a date range, it must be followed by the TO keyword with an ending value.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

DLVTIME

Specifies the time, or a period of time, when the transactions you want to work with were delivered to the application. If this is the start time of a time range, it must be followed by the `TO` keyword with an ending value.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE

PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

DOCREC

Requests that the document records are written to the EDIQUERY file. Valid values are:

Y Writes document records.

N Discards the document records. This is the default.

Note: N specifies that no data or image record are written to the query file.

For the format of these records, see “Transaction/Acknowledgment image data extract record layout” on page 320.

This keyword is used with **DOCUMENT DATA EXTRACT** command.

DOCSTAT

Indicates the processing status of the document.

For ENVELOPE AND SEND, the default value is **21**. Valid value is:

21 Send translated.

For REENVELOPE, the default values are **31**, **41**, **42**, and **43**. Valid values are:

29 Transaction detached -- send

30 Enveloped

31 Envelope error

41 Sent with errors

42 Send request error

43 Not sent network error

46 Send started status

48 Send requested

49 Sent to network

50 Accepted by network

51 Delivered by network

52 Purged by network

53 Recall requested

54 Recall request error

55 Recalled

61 Transaction accepted

62 Transaction rejected

63 Transaction accepted with errors

For TRANSLATE, the default value is **70**. For RETRANSLATE TO APPLICATION, the default value is **73**. Valid values for both are:

70 Received

72 Receive translated (RETRANSLATE TO APPLICATION only)

73 Receive translate error (RETRANSLATE TO APPLICATION only)

For UNPURGE TRANSACTIONS and REMOVE TRANSACTIONS, if no value is specified, all values are included in the selection criteria. Valid values are:

20 Send translate error

21 Send translated

29 Transaction detached -- send

30 Enveloped

31 Envelope error

41 Sent with errors

42 Send request error

43 Not sent network error

46 Send started status

48 Send requested

49 Sent to network

50 Accepted by network

51 Delivered by network

52 Purged by network

53 Recall requested

54 Recall request error

55 Recalled

61 Transaction accepted

62 Transaction rejected

63 Transaction accepted with errors

70 Received

71 Receive syntax error

72 Receive translated

73 Receive transaction error

74 Transaction detached–recv

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSLATE TO APPLICATION
UNPURGE

DOCUMENT

Specifies the document name for the input data. For application data, this field is required. For EDI and XML data, this field will override values extracted from the data. For EDI data, if you do not specify this keyword, the values in ST01 (X12) and UNH02 (EDIFACT) will be used to determine the value for this field. For XML data, if you do not specify this keyword, the root element will be used to determine the document name.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
TRANSFORM

DOTA1

Overrides the values in ISA14 and specifies wether a TA1 Interchange Acknowledgement is generated. Valid values are:

Y Generates a TA1 Interchange acknowledgment if the value found in ISA14 = 1. This is the default.

N Overrides the value found in ISA14 and does not generate a TA1 Interchange acknowledgment.

This keyword is used with the TRANSFORM command.

DUPCHECK

Indicates whether duplicate account/user ID and duplicate interchange ID/qualifier checks are performed when importing TPPROF members. This keyword is optional. Valid values are:

Y (or other)
Checks for duplicate IDs and ID qualifiers on TPPROF import. This is the default.

N Bypasses the duplicate checks on TPPROF import

Note: The import program checks the incoming TPPROF members to ensure that no duplicate accounts/user IDs or duplicate interchange ID qualifiers are imported. This is a time-consuming task and if you are confident that there are no duplicates in the import file, this keyword can be specified with a value of **N** to improve import performance. If duplicates are imported into the TPPROF and are referenced during translation, unpredictable results can occur.

This keyword is used only with the IMPORT command.

DUPENV

Indicates whether duplicate envelopes should be processed. Valid values are:

Y Processes duplicate envelopes. This is the default.

N Does not process duplicate envelopes. A condition code of 0005 is returned if you attempt to process a duplicate envelope.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE

RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
TRANSFORM

DYNSQL

Indicates whether dynamic SQL should be used for extracting the management report data. Dynamic SQL provides significant performance improvements if you maintain a lot of statistics. It enables DB2 to create an optimized plan specifically for your query. To use dynamic SQL, you must be authorized by your database administrator to access the appropriate views. Valid values are:

Y Uses dynamic SQL

N Uses static SQL. This is the default.

This keyword is used with the following commands:

NETWORK ACTIVITY DATA EXTRACT
TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

EENVDATE

Specifies the earliest date that transactions created during this run can be enveloped. The `Date mask` field in the language profile determines the format of the date code.

This keyword is used only with the TRANSLATE TO STANDARD command.

EIFORMAT

Indicates the requested export file record format. Valid values are:

TAGGED
Exports in tagged record format. This is the default.

FIXED Exports in fixed record format

This keyword is used the EXPORT command.

ENCODETARGET

Any supported encoding value can be specified. The ENCODETARGET value is used to generate the target data. This overrides any encoding value that is set in the data definition (for flat file data), or from the EncodeTarget map property.

This keyword is used with the TRANSFORM command.

ENVDATE

Specifies the date on which the transaction was enveloped. The maximum length is 10.

This keyword is used with the following commands:

ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO STANDARD
UNPURGE

ENVELOPE

Indicates if the EDI data should be enveloped (only used for outbound EDI). If N, the transactions are just added to the store for later enveloping. The default is Y.

This keyword is used with the TRANSFORM command.

ENVPRBREAK

Indicates whether a new interchange envelope or a new group envelope is started when the EDI standard envelope profile member name changes. Usually, the envelope profile provides the group envelope information. Use this keyword if your envelope profile provides interchange envelope information. Valid values are:

Y Starts a new interchange envelope

N Starts a new group envelope. This is the default.

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
REENVELOPE

REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

ENVTIME

The time at which the transaction was enveloped. The maximum length is eight.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
REMOVE TRANSACTIONS
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

ENVTYPE

For selecting transactions, indIcates the type of envelope used. Valid values are:

E UNB/UNZ

I ICS

T STX/END

U BG/EG

X ISA/IEA

0 Envelopes with no interchange header and trailer

For sending and receiving, indicates the type of receive issued. Valid values are:

E EDIFACT

I ICS or non-EDI file

T UN/TDI

U UCS

X X12 This is the default.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND SEND
RECEIVE AND TRANSLATE
RECVFILE AND SEND
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
SEND
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

EPURDATE

The date on which the transaction will be purged from the Document Store. The translator sets the default purge date when it adds the transaction to the Document Store. You can override the default length of time that the transaction can stay in the Document Store before being purged by using the `PURGINT` keyword when translating the transaction. The maximum length is 10.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY

PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

EXTEND

Indicates if the document record (D) should be appended with additional fields to produce an extended document record (DE). Valid values are:

Y Extend the record.

N Do not extend the record.

This keyword is used only with the DOCUMENT DATA EXTRACT command.

EXTENDC

Indicates whether the extended C record format is used when translating to application data with C and D records. Valid values are:

Y Uses the extended C record format

N Does not use the extended C record format. This is the default.

This keyword is used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND TRANSLATE
RETRANSLATE TO APPLICATION
TRANSLATE TO APPLICATION

EXTRACTED

To select previously extracted records. When used with command SAP STATUS EXTRACT, the default is N. When used with command SAP STATUS REMOVE, the default is Y.

Y extract only records already marked as extract

N extract only records NOT marked as extracted

A extract ALL records

FADELAY

Indicates whether functional acknowledgments are immediately enveloped and queued for sending, or are placed in the Document Store for enveloping and sending later. Valid values are:

Y Puts functional acknowledgments in the Document Store but does not envelope them

N Puts functional acknowledgments in the Document Store and also envelopes them to one of the following files:
- The file specified by the `FUNACKFILE` keyword, if present
- The TD queue from the network profile
- QDATA, QDATAU, or QDATAE (depending on envelope type)

This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
TRANSFORM

FILEID

Specifies the ddname of a file used to:
- Write data during an ENVELOPE operation
- Read data during a DEENVELOPE operation
- Send data during a SEND operation
- Write data during a RECEIVE operation

For outbound processing, if you do not specify this keyword, the value from the `Transaction data queue` field in the network profile is used as the ddname for the envelope file.

For inbound processing, if you do not specify this keyword, the value from the `Receive file name` field from the mailbox (requestor) profile is used as the ddname for the envelope file.

The maximum length is eight.

Notes:

1. On ENVELOPE and DEENVELOPE commands, all transactions are placed in this file. You should also specify the `NETID` keyword to make sure all transactions you select are for the same network.

2. FILEID contains the name of a TS queue. You should include this keyword to make sure that different applications running in the same CICS region do not envelope transactions to the same TS queue.
3. In CICS, WebSphere Data Interchange does not change the FILEID value to uppercase.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND SEND
ENVELOPE
ENVELOPE AND SEND
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND SEND
RECEIVE AND TRANSLATE
RECONSTRUCT
RECONSTRUCT AND SEND
RECVFILE AND SEND
REENVELOPE
REENVELOPE AND SEND
SEND
SENDFILE
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

FILTERMSGS

For data transformation processing. Filters messages in the list, if severity is less than 8. Maximum length is 80. Up to 11 individual messages can be filtered. See also IGNOREWARN and IGNOREINFO keywords.

This keyword is used with the TRANSFORM command.

FIXEDFILEID

Specifies the ddname of the file used for output during fixed-to-fixed translation processing. Data is written to the file during an ENVELOPE operation for fixed-to-fixed translation. The maximum length is eight.

If you do not specify this keyword, the ddname (based on the EDI standard ID) is the same as the Application file name in the target DF definition. To create a unique envelope file for each trading partner, specify the File Suffix field in the trading partner profile.

Notes:

1. On ENVELOPE commands, all transactions are placed in this file. You should also specify the NETID keyword to make sure all transactions you select are for the same network.

2. FIXEDFILEID contains the name of a TS queue. You should include this keyword to make sure that different applications running in the same CICS region do not envelope transactions to the same TS queue.
3. In CICS, WebSphere Data Interchange does not change the FIXEDFILEID value to uppercase.

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

FORCETEST

Indicates whether the deenvelope or translate-to-application processes should be forced to select a test usage/rule, regardless of the test indicator value in the envelope. This keyword is most useful when receiving test envelopes that do not have a test indicator (such as the UCS BG). In this case, you can use this keyword to force the translator to consider the envelopes for testing and only look for test usages/rules. Valid values are:

Y Forces the process to test mode and select only a test usage/rule if one is defined. If a test usage/rule is not found, an error is generated and the transaction is rejected.

If FORCETEST(**Y**) is used with the DEENVELOPE command, it must also be used on the TRANSLATE TO APPLICATION or RETRANSLATE TO APPLICATION commands to select the deenveloped transactions.

N Uses the test indicator from the envelope to determine which usage or rule to select. An envelope without a test indicator is always considered a production envelope. This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
TRANSLATE TO APPLICATION
RETRANSLATE TO APPLICATION

FORMAT

The ID of the data format associated with the transaction or transactions you want to select. The maximum length is 16.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

FUNACKFILE

The ddname of the file that you want to use for returning functional acknowledgments for the deenveloped transactions. You can use this keyword if you do not want to use the TD queue specified in the network profile. The maximum length is eight.

Note: In CICS, WebSphere Data Interchange does not change the `FUNACKFILE` value to uppercase and `FUNACKFILE` contains the name of a TS queue.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
TRANSFORM
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE

FUNACKP

Indicates whether transactions with functional acknowledgments pending should be selected. Valid values are:

Y Selects transactions for which a functional acknowledgment was requested but not received

N Selects transactions for which a functional acknowledgment was not requested or has already been received

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RECEIVE
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

FUNACKREQ

Indicates whether the functional acknowledgment envelope file is required. WebSphere Data Interchange will produce an error if unable to open it. Valid values are:

Y Requires the functional acknowledgment envelope file.

N Does not require the functional acknowledgment envelope file. This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE

GROUP

Indicates whether group data records are written to the EDIQUERY file. Valid values are:

Y Writes group data records.

N Discards group data records. This is the default.

For the format of these records, “Group data extract record layout” on page 315.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

GRPCTLLEN

Specifies the length of the group control number.

The generated control number is adjusted to the specified length. If the specified length is less than the default length, leading digits are removed. If the specified length is greater than the default length, then leading zeros are added.

If the specified length is greater than the maximum defined by the standard, then the maximum is used. If it is less than the minimum, then the minimum length is used. If the specified length is 0, then the default length is used.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

GRPCTLNO

Specifies the group control number assigned by the sender to identify the functional group. The maximum length is 14.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

HANDLE

Specifies the ID assigned by the system to a transaction when it is placed in the Document Store. To ensure uniqueness, the ID is a concatenation of the date, time, and a sequence number in format: YYYYMMDDHHMMSSnnnnnn

You can use this keyword to envelope a specific EDI document or all the documents whose time stamp falls within a given range. The system left-justifies and pads your entries. The `FROM` value is padded with 0s and the `TO` value is padded with 9s. To select transactions for the current date, use an asterisk (*).

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND

RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

HOLDFILE

The event log hold file name. The UNLOAD LOG ENTRIES command copies the non-archived event log entries into this file. The LOAD LOG ENTRIES command loads the records from this file into the event log. The maximum length is eight.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

HOLDTYPE

Indicates the event log hold file type. Used only for CICS and MQ (which is supported in both z/OS and CICS). Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name

TD Transient data queue

TM Temporary storage queue - main storage

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

In z/OS, the default is the ddname of the sequential file.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

IACCESS

Indicates how the interchange should be presented to the IEXIT program. Valid values are:

F Gives the interchange to the exit in a file. The interchange is written to the TD queue file, and then the IEXIT program is started.

M Gives the interchange to the exit in virtual storage. Applies only when `ITYPE` is **UE** (user exit).

IACCESS

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

IAREA

Specifies up to 16 bytes of information, the address of which is provided to the IEXIT program. Applies only to z/OS programs. In CICS, the address provided to the IEXIT program is the address of the utility control block. The maximum length is 16.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

ID

The name of a WebSphere Data Interchange profile such as Mailbox (REQPROF), Trading Partner (TPPROF), or Network (NETPROF). This must be a valid WebSphere Data Interchange profile ID. The maximum length is eight.

This keyword is used with the following commands:

QUERY PROFILE
DELETE PROFILE

IEXIT

The name of the program to receive control as each interchange is processed. See the ITYPE field description on 169 for the types of program you can use.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE

ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

IFCC

Specifies the condition codes that you want to override. You can specify up to 10 utility condition codes, separated by commas. The codes are checked by the WebSphere Data Interchange Utility and overridden with values from the `SETCC` keyword on a one-to-one basis. If you specify this keyword, you must specify the SETCC keyword.

This keyword can be used with any utility PERFORM command.

IGNOREBOM

This keyword prevents WebSphere Data Interchange from misinterpreting binary fields at the beginning of a data format record as a byte-order mark (BOM). A Byte Order Mark (BOM) is the character as a marker to indicate that text is encoded in UTF-8. Valid values are:

Y Ignore any byte-order mark, just treat as part of data.

N Process the byte-order mark as encoding information, not part of data. This is the default.

Note: This keyword only applies to data formats. It does not apply to EDI or XML, since these syntaxes cannot begin with binary information. If these bytes appear at the beginning of an EDI or XML file, the only valid way to interpret them would be as a byte-order mark.

This keyword is used with the TRANSFORM command.

IGNOREINFO

For data transformation processing. Filters all informational messages. The default is N.

This keyword is used with the TRANSFORM command.

IGNOREWARN

For data transformation. Filters all warning messages. The default is N.

This keyword is used with the TRANSFORM command.

IMAGE

Indicates whether image data records are written to the EDIQUERY file. Valid values are:

Y Writes image data records.

N Discards image data records. This is the default.

Images are always written as separate records. For the format of these records, “Transaction/Acknowledgment image data extract record layout” on page 320.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

INFILE

The input file containing the data. For z/OS, this is the ddname or MQSeries queue name. For CICS this is TS or TD queue, MQSeries queue, or VSAM data set.

This keyword is used only with the TRANSFORM command.

INMEMTRANS

Specifies the number of transactions held in memory before the database updates are attempted. This field is valid only if envelope level recovery (RECOVERY(**E**)) is in effect. Keeping transactions in storage delays the time when the database lock is obtained and reduces the length of time that the database lock is held. The more transactions kept in storage, the higher concurrency rate WebSphere Data Interchange can achieve. The amount of storage used for each transaction is approximately 2K. Valid values are **1** to **65535**. The default is **100**.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

INTCTLLEN

Specifies the length of the interchange control number.

The generated control number is adjusted to the specified length. If the specified length is less than the default length, leading digits are removed. If the specified length is greater than the default length, then leading zeros are added.

If the specified length is greater than the maximum defined by the standard, then the maximum is used. If it is less than the minimum, then the minimum length is used. If the specified length is 0, then the default length is used.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RECONSTRUCT
RECONSTRUCT AND SEND
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

INTCTLNO

Specifies the interchange control number assigned by the sender to identify the interchange data. The maximum length is 14.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS

PRINT TRANSACTION IMAGE
PURGE
QUERY
RECONSTRUCT
RECONSTRUCT AND SEND
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

INTERCHANGE

Indicates whether the interchange data record is written to the EDIQUERY file. Valid values are:

Y Writes interchange data records

N Discards interchange data records. This is the default.

For the format of these records, “Interchange data extract record layout” on page 314.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

INTID

The interchange sender/receiver ID of the trading partner as entered in the `Interchange ID` field of the trading partner profile.

This keyword is used only with the TRADING PARTNER PROFILE DATA EXTRACT command.

INTRECID

The interchange receiver ID (assigned by the receiver) that identifies the receiver to the sender. The maximum length is 35.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE

PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RECONSTRUCT
RECONSTRUCT AND SEND
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

INTRECQUAL

The interchange receiver ID qualifier that defines the type of the receiver ID. The maximum length is 4.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

INTSNDID

The interchange sender ID (assigned by the sender) that identifies the sender to the receiver. The maximum length is 35.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

INTSNDQUAL

The interchange sender ID qualifier that defines the type of the sender ID. The maximum length is 4.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS

RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

INTYPE

Indicates the type of file specified with the `INFILE` keyword. Used only for CICS and MQ (which is supported in both z/OS and CICS). For CICS, when you specify the `INFILE` keyword, you must also specify this keyword. Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name

TD Transient data queue

TM Temporary storage queue - main storage

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

VS VSAM database

This keyword is used only with the TRANSFORM command.

ITPBREAK

Indicates whether a new interchange envelope starts when the internal trading partner ID changes. Valid values are:

Y Always starts new interchange envelope. This is the default.

N Does not necessarily start a new interchange envelope.

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

ITYPE

Indicates the type of program specified in IEXIT. If you specify this keyword, you must also specify the IEXIT keyword. Valid values are:

PG A program that should be linked to using the EXEC CICS LINK command in CICS

UE A WebSphere Data Interchange user exit program defined in the User Exits profile

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

LASTTRXDATE

The date of the last transaction sent to or received from the trading partner.

This keyword is used only with the TRADING PARTNER PROFILE DATA EXTRACT command.

LEVEL

Specifies what information is dumped or how much detail is traced during processing.

This keyword is used with the following commands

GLB DUMP
GLB TRACE

For the GLB DUMP command, valid values are:

1 Dumps everything (default)

2 Dumps working storage

3 Dumps entire dataspace

10 Dumps common area

11 Dumps record index area

12 Dumps data area

For the GLB TRACE command, valid values are:

0 Ends trace (default)

1 Starts trace functions only

2 Starts trace functions and subroutines

3 Starts trace functions, subroutines, and subroutine key points

LOGAEID

The associated entry ID in the event log. This keyword can be used with the TO keyword to specify a range of associated entry IDs for selecting event log entries. The maximum length is 40.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

LOGDATE

The event log date. This keyword can be used with the TO keyword to specify a range of dates for selecting event log entries. The maximum length is eight.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

LOGFAMSG

This keyword generates an alert when a Functional Acknowledgement is overdue that is written to the PRTFILE, XMLPRT, and ADFPRT. The user defined message can be up to 10 characters.

This keyword is used only with the TRANSACTION DATA EXTRACT command.

LOGFORM

The event log format ID. This keyword can be used with the TO keyword to specify a range of format IDs for selecting event log entries. The maximum length is 16.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

LOGTIME

The event log time. This keyword can be used with the TO keyword to specify a range of times for selecting event log entries. The length is six.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

LOGUSER

The event log user ID. This keyword can be used with the `TO` keyword to specify a range of user IDs for selecting event log entries. The maximum length is eight.

This keyword is used with the following commands:

LOAD LOG ENTRIES
REMOVE LOG ENTRIES
UNLOAD LOG ENTRIES

MAPID

The map name. For the TRANSFORM command, the value in this field overrides the map name specified in data transformation rules.

This keyword is used with the following commands:

DELETE USAGE
TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT
TRANSFORM

MAXRUNTIME

The maximum time in minutes that the REMOVE TRANSACTIONS process can run. Once the specified time is reached, the REMOVE TRANSACTIONS process stops even if it has not completed. When the REMOVE TRANSACTIONS process is running, other WebSphere Data Interchange processes are prevented from running. Setting this keyword and value is useful when the REMOVE TRANSACTION process is to run standalone for a limited time. The default value is **0** (no maximum run time).

This keyword is used only with the REMOVE TRANSACTIONS command.

MEMBER

The name of a WebSphere Data Interchange profile member. The maximum length is dependent upon the type of profile but cannot exceed 35.

This keyword is used with the following commands:

DELETE PROFILE
QUERY PROFILE

REPORT CONTINUOUS RECEIVE STATUS
START CONTINUOUS RECEIVE
STOP CONTINUOUS RECEIVE

MERGED

Indicates whether the transaction image is printed with a new line for each segment, and merged with the functional acknowledgment image. Valid values are:

Y Prints each segment on a new line

N Does not print each segment on a new line

This keyword is used only with the PRINT TRANSACTION IMAGE command.

MRREQID

A requestor ID used to associate management reporting statistics with DEENVELOPE processing instead of RECEIVE processing. You can use this keyword if you receive interchanges without using WebSphere Data Interchange , and you want to keep management reporting receive statistics on those interchanges. Do not use this keyword if management reporting statistics were created when the interchange was received (default). Use this keyword only in exceptional situations.

This keyword is used with the following commands§

DEENVELOPE
DEENVELOPE AND TRANSLATE
RECEIVE AND DEENVELOPE

MSGUCLASS

An override message user class for a send or receive type command. If you specify this keyword, its value overrides the value specified in the mailbox (requestor) profile member identified by the `REQID` keyword. The maximum length is eight.

This keyword is used with the following commands:

ENVELOPE AND SEND
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND SEND
RECEIVE AND TRANSLATE
RECONSTRUCT AND SEND
RECVFILE AND SEND
REENVELOPE AND SEND
SEND
SENDFILE
TRANSLATE AND SEND

MULTIDOCS

Indicates whether the XML input file contains multiple documents. Valid values are:

Y Multiple documents. The input message must be in EBCDIC format and each document must begin with an XML declaration (`<?xml...`).

N One document. This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE

NETACKP

Indicates whether to select transactions for which network acknowledgments are pending. Valid values are:

Y Selects transactions for which a network acknowledgment was requested but not received

N Selects transactions for which a network acknowledgment is not pending or was not requested

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

NETID

The network ID as entered in the network profile. The maximum length is eight.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
NETWORK ACTIVITY DATA EXTRACT
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE
UPDATE STATUS

NETNAME

The name of the network as entered in the `Network name` field of the network profile. This field is case sensitive. The maximum length is 30.

This keyword is used only with the NETWORK ACTIVITY DATA EXTRACT command.

NETSTAT

Indicates the network status of a transaction for which a send has been requested. Valid values are:

30 Enveloped

31 Envelope error

41 Sent with errors

42 Send request error

43 Not sent net error

46 Send started

48 Send requested

49 Sent to network

50 Accepted by network

51 Delivered by network

52 Purged by network

53 Recall requested

54 Recall request error

55 Recalled

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RECEIVE AND SEND
RECVFILE AND SEND
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
SEND
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD
UNPURGE

NEWAPPLID

The application ID used to change to another application when loading event log entries into an event log. The maximum length is eight.

This keyword is used only with the LOAD LOG ENTRIES command.

NOMSG

Indicates whether extraneous messages should appear in the print file. Applies only to EXPORT and IMPORT commands, and is normally used when large numbers of records are being exported or imported. For example, it can be used with export-all (exporting after building a control file using program EDIXPEA) or import-all (using a control file with category 9).

Y Suppresses the following messages from appearing in the print file:

`EI0048, EI0049, EI0062, FF0512, FF0514`

Also suppresses certain error conditions. For example, if in your command file you specify that you want to export CONTRECV members, and no Continuous Receive (CONTRECV) profile exists on your system, this error is ignored. Also, if you specify certain associated objects that do not exist, this error is also ignored.

N Does not suppress error messages or conditions. This is the default.

This keyword is used with the following commands:

EXPORT
IMPORT

NUMDELS

Specifies the number of database deletions performed before WebSphere Data Interchange issues a database commit. DB2 includes a site-dependent value (in `NUMLKUS`) that controls the maximum number of page locks that a single REMOVE TRANSACTIONS process can apply. When the REMOVE TRANSACTIONS process is not running standalone, page locks are obtained against the database. If the `NUMDELS` value is too high, DB2 can stop the REMOVE TRANSACTIONS process because the value in `NUMLKUS` has been exceeded. If this happens, use a smaller number in `NUMDELS` so that WebSphere Data Interchange issues database commits more frequently, which will release page locks more quickly. The default is **100** and the maximum is **1000**. If a value greater than 1000 is entered, the default of 100 is used.

This keyword is used with the following commands:

REMOVE LOG ENTRIES
REMOVE STATISTICS
REMOVE RANSACTIONS
RESET STATISTICS
UNLOAD LOG ENTRIES

NUMUPDTS

Specifies the number of database updates performed before WebSphere Data Interchange issues a database commit. If you are experiencing timeouts using a command that includes this parameter, you can reduce the amount of time that the command holds page locks by forcing more frequent COMMITs. The default for NUMUPDTS is **50**. If you want more frequent COMMITS performed, specify a number smaller than 50.

If you are trying to improve the performance of a command that includes this parameter, you can specify a value larger than 50. This forces WebSphere Data Interchange to COMMIT less frequently and should speed up processing.

This keyword is used only with the UPDATE STATISTICS command.

ONELOGICAPP

Indicates whether the `APPFILE` in the current WHERE clause and all `APPFILEs` in proceeding WHERE clauses are considered one logical file. When chosen, envelope breaks are avoided between `APPFILE` processing. You can use this keyword to avoid `APPFILE` switching that causes envelope breaks when processing multiple raw data files by placing multiple raw data files in the same envelope without delaying enveloping. If the `FILEID` keyword is specified in combination with this keyword in multiple WHERE clauses, the value in the first `FILEID` found is used to envelope the data. You can also use this option with C and D record formats. Valid values are:

Y Treats multiple application files as one logical file.

N Processes each file independently; envelope breaks occur at the end of each application file. This is the default.

This keyword is used with the following commands:

TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

ONEMSG

Indicates whether all MQSeries messages are read from an MQSeries queue or only one message is read from the queue. The queue can be either a receive file or an application send file. Applies only to MQ. An MQSeries message is defined as a logical set of MQ records with the same MSGID. This keyword also controls MQ message descriptor propagation when you can set it on the TRANSLATE AND SEND and RECEIVE AND TRANSLATE commands where the application file and the send or receive file are both MQ queues. If you want MQ message descriptors to propagate during processing, set this keyword to **Y**.

Indicates whether all messages or documents in the output data files are written as a single MQSeries queue message or if they are separated into different MQSeries queue messages by a SEND or SENDFILE command. Set this keyword to Y to direct

WebSphere Data Interchange to add a separator between logical messages or documents and N to not include separators. N is the default value. Because output data files will have a separator record inserted when ONEMSG(Y) is used, it is not advisable to use it if the outfile will not be processed by a SEND or SENDFILE command.

When used in conjunction with ONEMSG(Y) in the TRANSFORM command, and when the value Y is set in this keyword, the logical messages/documents in the output file is written as different MQSeries queue messages. When the value is N the output file is written as a single MQSeries queue message. Any separator records are bypassed.

Valid values are:

Y Apply ONEMSG Rules for Reads and Writes from an MQSeries queue.

N Do not Apply ONEMSG Rules for Reads and Writes from an MQSeries queue. This is the default.

This keyword is used with the following commands:

RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
SEND
SENDFILE
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

OPTRECS

Indicates which optional records are created during translation. **I** is valid only for the TRANSLATE TO STANDARD command (delayed enveloping). **Q** is not valid for the TRANSLATE or RETRANSLATE TO APPLICATION commands. The maximum length is 5. Valid values are:

E Envelope header

G Group header

I Information

Q Queuing

T Transaction header

For more information on these records, “Optional records” on page 301.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE

ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
RETRANSLATE TO APPLICATION
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

OUTFILE

The ddname of an output file (or the name of a TS queue or TD queue). For CICS, when you specify this keyword, you must also specify the OUTTYPE keyword. For the **TRANSFORM** command, the value in this field overrides any output file names generated during processing, such as the filename from the trading partner profile. The maximum length is eight.

This keyword is used with the following commands:

QUERY PROFILE
SAP STATUS EXTRACT
TRANSFORM

OUTFORMAT

Indicates the format in which the output is written. Valid values are:

F Fixed

N Native

T Tagged

This keyword is used only with the QUERY PROFILE command.

OUTLEN

Specifies the maximum record length for the output data.

This keyword is used only with the TRANSFORM command.

OUTTYPE

Indicates the file type of `OUTFILE`. Applies only for CICS and MQ (which is supported in both CICS and z/OS). For CICS, when you specify the `OUTFILE` keyword, you must specify this keyword. Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name.

TD Transient data queue.

TM Temporary storage queue - main storage.

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

VS VSAM data set.

This keyword is used with the following commands:

QUERY PROFILE
SAP STATUS EXTRACT
TRANSFORM

For z/OS, if you do not specify this keyword, this field is ignored and the ddname of a sequential file is used. For CICS, the default is **TS**.

PAGE

For the TRANSFORM command, this indicates whether WDI should use an intermediate file EDIWORK when it creates large output files. Valid values when using the TRANSFORM command:

Y Use file EDIWORK as a temporary work file to create large output files.

N Do not use file EDIWORK when creating output files. Instead, the output from the TRANSFORM command is held in memory until WDI is ready to write it to the output file. For large output files, this may use significantly more memory than PAGE(Y).

For other commands, this indicates whether pageable translation should be enabled. For more information about pageable translation, see *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

Valid values for Pageable Translation are:

Y Enables Pageable Translation

N Disables Pageable Translation

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
RECONSTRUCT AND SEND
REENVELOPE

REENVELOPE AND SEND
RETRANSLATE TO APPLICATION
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

PAGETHRESHOLD

Overrides the threshold that specifies when the AMM subtree containing the repeating data is paged out of memory. The default threshold is 1000. 0 turns paging off.

Notes:

1. This feature is not available when in CICS. The primary reason for this is the use of a sequential file as the EDIPAGE file and handling of sequential files within CICS. A design point is also that Very Large Transactions, as those suited for the Pageable AMM feature, are more than likely not acceptable for transaction oriented CICS processing.
2. The page file is named EDIPAGE. If an error occurs opening the page file or writing to the page file, transformation continues normally, though a severity 4 warning message is logged.

PARSEFILE

This keyword specifies that the data is to be parsed directly from the input file. Using this keyword can significantly reduce the memory needed for processing a large input file. Without this feature, WebSphere Data Interchange reads the document into a buffer, so if a 500 MB document is being processed, then WebSphere Data Interchange uses at least 500 MB to parse the data. With this option, the WebSphere Data Interchange parsing reads from the file and requires less memory.

Valid values are:

Y Species to parse from file.

N Specifies to parse from buffer. This is the default.

Notes:

1. PARSEFILE is only valid for input data, and is not valid in CICS.
2. PARSEFILE(Y) should only be specified when the input file contains one and only one document. More than one document in the file results in a fatal error in the parser.

This keyword is used with the TRANSFORM command.

PRIORTO

Specifies the date before which all statistics will be deleted.

This keyword is used with the following commands:

REMOVE STATISTICS
RESET STATISTICS
SAP STATUS REMOVE

PURGINT

Specifies the number of days that a transaction remains in the Document Store before being marked for purging. If you do not specify this keyword and value, or if you specify a value of **0**, the WebSphere Data Interchange Utility uses **30** days as the default. The maximum value is **9999**. You can use a negative value to indicate that a transaction's store time expired on a date in the past. For example, if you specify `PURGINT with a value of` **-2**, the purge date is set at two days ago. The minimum value is **-999**. If a functional or network acknowledgment is pending when the store time expires, the transaction retains its current store status until the acknowledgment is no longer pending.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
RETRANSLATE TO APPLICATION
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

RAWDATA

Indicates whether you want the data translated from EDI standard format to raw format, or to C and D records. Valid values are:

Y Raw data format

N C and D record format

For more information about these record formats, see the . For an explanation of the file to which data will be written, *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

This keyword has this meaning when used with the following commands:

DEENVELOPE AND TRANSLATE
RECEIVE AND TRANSLATE
RETRANSLATE TO APPLICATION
SENDFILE
TRANSLATE TO APPLICATION

For Fixed-to-Fixed translation, this keyword indicates whether the output data should be written in raw data format or in the C and D record format. For a description of C and D records, Chapter 5, “Utility records format,” on page 285.

This keyword has this meaning when used with the following commands:

ENVELOPE
ENVELOPE AND SEND
RECEIVE AND SEND
RECONSTRUCT
RECONSTRUCT AND SEND
REENVELOPE
REENVELOPE AND SEND
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

RAWFMTID

Applies only for raw data. Specifies the format ID of the application data file. The maximum length is 16.

This keyword is used with the following commands:

TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

RAWTEST

Applies only to raw data. Indicates to the receiver whether the transaction is for testing. Valid values are:

Y Test transactions. All the transactions are test transactions, and a test usage/rule should be used, if it exists. If a test usage/rule does not exist, the production usage/rule should be used. Same as RAWUSAGE(**T**).

N Production transactions. All the transactions are production transactions and only the production usage/rule should be used. Same as RAWUSAGE(**P**).

U

This is the default.

Either test or production transactions. WebSphere Data Interchange determines the status of the transaction based on the presence of an active test usage or rule. If an active test usage or rule exists, the transaction is considered a test transaction. If an active test usage or rule does not exist, the transaction is considered a production transaction. Same as RAWUSAGE(**U**).

Note: This keyword is being replaced by RAWUSAGE. The RAWUSAGE value overrides RAWTEST if both values are specified. RAWTEST is provided for transitional purposes only, and will be removed in a future release.

The translator uses this value to set the test indicator in the interchange header (0035 for EDIFACT, I14 for X12). For C and D records, the TESTIND field of the control record serves the same purpose as this keyword.

This keyword is used with the following commands:

TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

RAWUSAGE

Applies only for raw data. Indicates to the receiver whether the transaction is for production, testing, or information. Valid values are:

P Production transactions. All of the transactions are production transactions and only the production usage or rule should be used. This is the default.

T Test transactions. All of the transactions are test transactions, and a test usage or rule should be used. If a test usage or rule does not exist, the production usage or rule should be used.

I Information transaction. An information usage/rule is used if one exists. If one does not exist, a production usage or rule should be used. If a production usage or rule does not exist, an error occurs.

U Either test or production transactions. WebSphere Data Interchange determines the status of the transaction based on the presence of an active test usage or rule. If an active test usage/rule exists, the transaction is considered a test transaction. If an active test usage or rule does not exist, the transaction is considered a production transaction.

The translator uses this value to set the test indicator in the interchange header (0035 for EDIFACT, I14 for X12). For C and D records, the TESTIND field of the control record serves the same purpose as this keyword.

This keyword is used with the following commands:

TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

RECEIVEACKDATA

Indicates whether detailed receive acknowledgment data is written to the EDIQUERY file. Valid values are:

Y Writes detailed acknowledgment data.

N Discards detailed acknowledgment data. This is the default.

Detailed acknowledgment data includes interchange, group, and transaction data for each acknowledgment transaction. This information is concatenated with the group and transaction records to which it applies. For the format of these records, “Transaction/Acknowledgment image data extract record layout” on page 320.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

RECEIVEACKIMAGE

Indicates whether the receive acknowledgment image is written to the EDIQUERY file. Valid values are:

Y Writes receive acknowledgments.

N Discards receive acknowledgments. This is the default.

Images are always written as separate records. For the format of these records, “Transaction/Acknowledgment image data extract record layout” on page 320.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

RECOVBAD

Indicates whether the translation process will try to recover from a bad EDI standard data flag. Valid values are:

Y The translation process tries to recover from bad EDI standard data. The translator checks for the standard interchange headers when a segment terminator is missing from a particular standard segment. The translator attempts to reset the delimiters and check the current segment/record for the segment terminator.

N The translation process does not try to recover. This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE

RECOVERY

Indicates the unit of work. When this keyword is used on the PERFORM TRANSFORM command, RECOVERY(E) will cause all transactions in an interchange to be discarded if any of the transactions fails. If RECOVERY(T) is specified, the successful transactions from the interchange will be processed and only the invalid transactions will be discarded. The default is environmentally dependent. Valid values are:

E Issues a database commit after each envelope. This is the default for CICS.

T Issues a database commit after each transaction. This is the default for z/OS.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
REENVELOPE
REENVELOPE AND SEND
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

REQID

The requestor ID as entered in the mailbox (requestor) profile. Since each WHERE clause can contain only one requestor ID, you must add a WHERE clause for each requestor ID. A send is issued for each requestor ID for which data was queued. The maximum length is 16.

This keyword is used with the following commands:

CLOSE MAILBOX
DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE AND SEND
NETWORK ACTIVITY DATA EXTRACT
PROCESS NETWORK ACKS
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND SEND
RECEIVE AND TRANSLATE
RECVFILE AND SEND

REENVELOPE AND SEND
SEND
SENDFILE
TRANSLATE AND SEND
UPDATE STATUS

REQTP

The trading partner associated with a given requestor ID for use with direct connection networks such as point-to-point. This keyword is used to eliminate the need to specify a separate requestor ID for each trading partner.

This keyword is used only with the ENVELOPE AND SEND command.

RESET

Indicates whether the output file is reset to receive new data or the new data is appended to the data already in the output file. Valid values are:

Y Resets the output file

N Appends data to the output file

This keyword is used with the following commands

GLB DUMP
GLB TRACE

SAPFILE

The ddname of an output file (or the name of a TS queue or TD queue). For CICS, when you specify this keyword, you must also specify the SAPTYPE keyword. For the TRANSFORM command the system will write the SAP status data to the file specified by the SAPFILE/SAPTYPE keywords for CICS or to the file specified by the SAPFILE keyword for other platforms. The system will also write the data to the DB; each record will be marked as extracted. The maximum length is eight.

This keyword is used with the TRANSFORM command.

SAPTYPE

Indicates the file type of SAPFILE. Applies only for CICS and MQ (which is supported in both CICS and z/OS). For CICS, when you specify the SAPFILE keyword, you must specify this keyword. Valid values are:

MQ WebSphere Data Interchange MQSeries queue profile member name.

TD Transient data queue.

TM Temporary storage queue - main storage.

TS Temporary storage queue - auxiliary storage. This is the default for CICS.

VS VSAM data set.

This keyword is used with the TRANSFORM command.

SAPSTAT

Specifies the SAP status value to extract, or when used with the TO keyword, the range of values to extract. Valid values are 04–22. The default is all status.

This keyword is used with the following commands:

RECEIVE AND SEND
SAP STATUS EXTRACT
SAP STATUS REMOVE

SAPUPDT

Indicates whether SAP status tracking is required. Valid values are:

Y Tracks SAP status and writes status records.

N Does not track SAP status or write status records. This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND SEND
RECVFILE AND SEND
REENVELOPE
REENVELOPE AND SEND
SEND
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO STANDARD

SCRIPT

Specifies the value that can be used by communication software to identify a set of instructions to follow when processing requests for services. This set of instructions would be part of the communication software package and not part of WebSphere Data Interchange . The maximum length is eight.

This keyword is used with the following commands:

ENVELOPE AND SEND
RECEIVE AND SEND
RECONSTRUCT AND SEND
RECVFILE AND SEND
REENVELOPE AND SEND
RESTART SEND
SEND
SENDFILE
TRANSLATE AND SEND

SEGMENTED

Indicates whether the image is printed with each segment starting on a new line. Valid values are:

Y Starts a new line for each segment

N Does not start a new line for each segment

This keyword is used with the following commands:

PRINT ACKNOWLEDGMENT IMAGE
PRINT TRANSACTION IMAGE

SENDACKDATA

Indicates whether detailed send acknowledgment data is written to the EDIQUERY file. Valid values are:

Y Writes detailed acknowledgment data.

N Discards detailed acknowledgment data. This is the default.

Detailed acknowledgment data includes interchange, group, and transaction data for each acknowledgment transaction. This information is concatenated with the group and transaction record to which it applies. This keyword is ignored unless `GROUP` or `TRANSACTION` is set to **Y**. For the format of these records, see “Transaction/ Acknowledgment image data extract record layout” on page 320.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

SENDACKIMAGE

Indicates whether the send acknowledgment image is written to the EDIQUERY file. Valid values are:

Y Writes the generated acknowledgment record.

N Discards the generated acknowledgment record. This is the default.

Images are always written as separate records. For the format of these records, see “Transaction/Acknowledgment image data extract record layout” on page 320.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

SEQNUM

Indicates whether network profile member sequence numbers should be incremented during send processing. This is an optional keyword. Specifying a value of **N** turns off network sequence numbers and saves associated overhead. Valid values are:

Y Increments network profile member sequence numbers. This is the default.

N Does not increment network profile member sequence numbers.

This keyword is used with the following commands:

ENVELOPE AND SEND
REENVELOPE AND SEND
RESTART SEND
SEND
SENDFILE
TRANSLATE AND SEND

SERVICESEGVAL

Indicates at which level the service segments should be validated. The service segments are the segments used when a transaction is enveloped (ISA, GS, ST, UNB, UNH, UNT, and so on). If you do not specify this keyword, no validation occurs. Valid values are:

1 Validates the service segments for syntax only. This includes checking for mandatory data that is missing, as well as data elements that are too large or too small.

2 In addition to level 1 checking, validates the service segment date and time data elements according to their types, and if a validation table has been specified, also checks the value of the data element.

Validation errors that occur during send processing will terminate processing. Validation errors that occur during receive processing cause the interchange/group/transaction with the error to be skipped (not processed) but processing continues.

For Data Transformation processing the service segment validation is accomplished using WebSphere Data Interchange validation mapping. The validation maps are executed in addition to any inbound validation maps specified on the data transformation mapping rule. The keyword SERVICESEGVAL on the PERFORM TRANSFORM command will identify the level of service segment validation.

```
&WDI_E99AENV_VAL - UN/EDIFACT based on E99A service segments and code lists
&WDI_UCSENV_VAL - UCS based on UCS 4050 service segments and code lists
&WDI_X44ENV_VAL - X12 based on X12 4040 service segments and code lists
```

The validation map names cannot change with WebSphere Data Interchange V3.3 and are selected based on the inbound interchange or envelope type. Although the shipped validation maps can be modified, it is recommended that customers who want to modify the mapping, make a copy of the validation map and update the mapping to add a DIMAPSWITCH command to switch to the copied validation mapping.

The following translation tables are used within the mapping:

```
&WDI_E99AENV_VAL
ETEXT00A - For functional acknowledgment error code translation to error
message text included with WDI error message UT0033
&WDI_UCSENV_VAL
A1DEC - For error code translation to functional acknowledgment error code
A1TEXT - For functional acknowledgment error code translation to error
message text included with WDI error message UT0033
&WDI_X44ENV_VAL
TA1INC - For error code translation to TA1 note code
TA1MT - For TA1 note code translation to error message text included with
WDI error message UT0033.
```

A new data transformation property SERVSEGVAL has also been added to be used with mapping commands GetProperty().

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
RECEIVE AND DEENVELOPE
REENVELOPE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

SETCC

Specifies the condition codes used by the WebSphere Data Interchange Utility to override the utility condition codes specified in the IFCC keyword. You can have up to 10 override condition codes separated by commas. If you specify the IFCC keyword, you must specify this keyword. If this keyword is omitted, all condition codes specified on the IFCC keyword are overridden to zero (0). If a particular code is omitted in the SETCC keyword, the related condition code in the IFCC keyword is overridden to zero (0). For more information, see “Overriding utility condition codes” on page 23.

This keyword can be used with any utility PERFORM command.

SNDDATE

The date of the previous send request for the transaction. The maximum length is 10.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

SNDTIME

The time of the previous send request for the transaction. The maximum length is eight.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS

TRANSACTION DATA EXTRACT
TRANSFORM
UNPURGE

SOURCEENCODE

Any supported encoding value can be specified. The SOURCEENCODE keywords is used with the PERFORM TRANSFORM. The SOURCEENCODE value is used to interpret the source data. This overrides any encoding value that is set in the data definition or the data. These values are typically set using the encoding= value in the XML data or the syntax id in EDIFACT data.

The SOURCEENCODE keyword has a special substitution value that enables WebSphere Data Interchange to use the value from the MQ header. If the SOURCEENCODE value is set to MQCCSID, the parser converts this value as follows:

- If the incoming message has an RFH2 header, it gets the CodedCharSetId value from the RFH2 header.
- If the incoming message has no RFH2 header, but has an MQMD header, it gets the CodedCharSetId value from the MQMD header.
- If a CodedCharSetId value is found in either the MQMD or RFH2 header, the parser converts the CCSID to an encoding name as follows:
  1. The parser looks up the value in the translate table CCS2ENC.

     Note: This value is an integer value. The parser needs to convert the integer value to a character string for the table lookup.

     For example, the value for RFH2 CCSID value is 1208. The translate table that the value 1208 is mapped to the Unicode value of UTF-8, then the value UTF-8 is substituted for the SOURCEENCODE value.
  2. If the value is not found in step 1, the encoding name “ibm-nnnn” is used, where “nnnn” is the CCSID.

     For example, if the CCSID value is 1208 and it is not in the CCS2ENC table, the encoding name ibm-1208 is used.
  3. The converted value is used as the SOURCEENCODE value and an informational message will be issued to indicate which encoding name was used.
- If the message does not have an RFH2 or MQMD header, a warning message is issued and the source encoding is processed as if the SOURCEENCODE value was not specified, that is, it is detected from the data or the default encoding.

This keyword is used on the TRANSFORM command.

STANDALONE

Indicates whether the Remove Transactions process should operate in contention with other WebSphere Data Interchange processes. In the DB2 environment, this obtains exclusive high-level table locks which improves performance of the Remove Transactions process since WebSphere Data Interchange does not need to obtain lower-level page locks. Valid values are:

Y The Remove Transactions process runs alone. Does not contend with any other WebSphere Data Interchange process.Run the Remove Transaction process in this mode if possible.

N The Remove Transactions process does not run alone. Contends with other WebSphere Data Interchange processes. High-level table locks are not obtained. This is the default.

This keyword is used only with the REMOVE TRANSACTIONS command.

STDDESC

The description of the EDI standard as entered in the standards database. This field is case sensitive.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

STDID

The EDI standard ID as entered in the standards database.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

STDLV

The level of the EDI standard, for example, R1 for release 1.

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

STDTRID

The transaction ID of the EDI standard as specified by the standard, such as 850 for an X12 purchase order. The maximum length is eight.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT

HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

STDVR

The version of the EDI standard, for example, V3 for version 3.

This keyword is used with the following commands

TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

STSTAT

Indicates the status of the transaction in the Document Store. Valid values are:

0 Active

1 Held

3 Purge-date expired

4 Purge-user request

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY

PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RELEASE
REMOVE TRANSACTIONS
TRANSACTION DATA EXTRACT
UNPURGE

SYNTAX

Specifies the syntax type for the input data. This is a required field. Valid values are:

D Application data

E EDI data

X XML data

This keyword is with the following commands:

DOCUMENT DATA EXTRACT
TRANSFORM

TESTMODE

Indicates whether the transactions are test, information, or production transactions. Valid values are:

Y Test or information transactions

N Production transactions

This keyword is used with the following commands:

TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

THANDF

The THANDF keyword specifies a logical file name which contains a list of thandles to be processed by any PERFORM command which takes a thandle as an argument. The file should contain a list of thandles seperated by spaces.

Note: This keyword is only used for submitting Document Store commands from the WebSphere Data InterchangeClient.

TPID

The internal trading partner ID used by an application as entered in the trading partner send usage/rule. The maximum length is 35.

This keyword has this meaning with the following commands:

DELETE USAGE
ENVELOPE
ENVELOPE AND SEND
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

This keyword provides the default internal trading partner ID value, if the data format does not define a field that contains this value, or if a field is defined but contains all blanks. This also becomes the default ID value if the internal trading partner ID is blank in the C record.

This keyword has this meaning with the following commands:

TRANSLATE TO STANDARD
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND

TPNICKN

The trading partner nickname for a trading partner profile member. The maximum length is 16. TPNICKN has two distinct uses:

- This keyword specifies a trading partner to receive data from, or a trading partner nickname to use when selecting Document Store or management reporting data.

- For the SEND and TRANSLATE AND SEND commands, this keyword is used the same as the keyword `TPNICKNESEND`. This means that the fields in the specified trading partner profile member are used for network override options (such as network charges), and will override the same fields in the mailbox (requestor) profile.

  Note: When this keyword is used with the SEND and TRANSLATE AND SEND commands to send EDI data, the receiver's mailbox is not specified because that information is contained within the data being sent.

This keyword is used with the following commands:

DELETE USAGE
DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
RECEIVE
RECEIVE AND DEENVELOPE
RECEIVE AND SEND
RECONSTRUCT
RECONSTRUCT AND SEND
RECVFILE AND SEND
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
SEND
SENDFILE
TRADING PARTNER CAPABILITY DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

TPNICKNESEND

Specifies a trading partner profile member to use for network override options (such as network charges), that will override the same fields in the mailbox (requestor) profile.

Note: When this keyword is used with the SEND and TRANSLATE AND SEND commands to send EDI data, the receiver's mailbox is not specified because that information is contained within the data being sent.

This keyword is used with the following commands:

ENVELOPE AND SEND
REENVELOPE AND SEND

TRACELEVEL

Indicates the level of tracing done during the transform process. For z/OS, trace data will be written to ddname **EDIDTTRC**. For CICS, trace data will be written to the TD queue defined for EDI standard output. If required you can change the TD queue to a TS queue. For AIX and Windows platforms the trace data will be written to the file defined by the environment variable **EDIDTTRC**. You can set this using `export` command on AIX platforms or the `set` command on Windows platforms. For example:

```
export EDIDTTRC=trace.out
```

or

```
set EDIDTTRC=trace.out
```

The value consists of a series of C*n* values which represent the component and trace level for the component. The valid values for the component ID are:

A All nodes

D Deenveloper node

E Enveloper node

M Message broker

P Parsers

R Rules node

T Transformation node

V Validation node

The valid values for the component tracing level are:

0 All trace messages are ignored.

1 Normal tracing. Only the first 256 byes of data in the buffer are written to the trace file.

-1 End user data tracing.

2 Extended tracing. The entire contents of the buffer is written to the trace file.

3 Utility function tracing. Includes all the tracing done at level **2** plus additional tracing for some frequently called internal utility functions.

For example, a value of D1 V2 R2 would mean the deenveloper node (**D**) is set for normal tracing (**1**), the validation (**V**) and rules (**R**) nodes are set for extended tracing (**2**). Tracing is normally turned off except during problem determination. Activating tracing can negatively impact performance.

This keyword is used only with the **TRANSFORM** command.

TRANSACTION

Indicates whether the transaction data record is written to the EDIQUERY file. Valid values are:

Y Writes transaction data records.

N Discards transaction data records. This is the default.

For the format of these records, see “Transaction/Acknowledgment image data extract record layout” on page 320.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
TRANSACTION DATA EXTRACT

TRERLVL

Indicates the maximum translation error level for the transactions you want to select. You can use this keyword to envelope only EDI documents that are error-free. Valid values are:

0 No errors

1 Data element errors

2 Data element and segment errors

3 Severe errors

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE

QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
UNPURGE

TRKFILE

Specifies that a tracking file to be used. If you specify **TRKFILE(Y)** and specify a tracking file in your command file your optional records will go into that tracking file instead of the application file.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND SEND

TRXCTLLEN

Specifies the length of the transaction control number.

The generated control number is adjusted to the specified length. If the specified length is less than the default length, leading digits are removed. If the specified length is greater than the default length, then leading zeros are added.

If the specified length is greater than the maximum defined by the standard, then the maximum is used. If it is less than the minimum, then the minimum length is used. If the specified length is 0, then the default length is used.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS

RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

TRXCTLNO

The transaction set control number assigned by the sender to identify the transaction set to the sender. When combined with the sender ID, it identifies the transaction set to the receiver. The maximum length is 14.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

TRXDATE

The date the transaction was added to the Document Store. This is also the date on which the transaction was translated to EDI standard format. The maximum length is 10.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT

HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

TRXSTAT

Indicates the processing status of the transaction.

For **ENVELOPE AND SEND**, the default value is **21**. Valid value is:

21 Send translated.

For **REENVELOPE**, the default values are **31**, **41**, **42**, and **43**. Valid values are:

29 Transaction detached -- send

30 Enveloped

31 Envelope error

41 Sent with errors

42 Send request error

43 Not sent network error

46 Send started status

48 Send requested

49 Sent to network

50 Accepted by network

51 Delivered by network

52 Purged by network

53 Recall requested

54 Recall request error

55 Recalled

61 Transaction accepted

62 Transaction rejected

63 Transaction accepted with errors

For **TRANSLATE**, the default value is **70**. For **RETRANSLATE TO APPLICATION**, the default value is **73**. Valid values for both are:

70 Received

72 Receive translated (RETRANSLATE TO APPLICATION only)

73 Receive translate error (RETRANSLATE TO APPLICATION only)

For **UNPURGE TRANSACTIONS** and **REMOVE TRANSACTIONS**, if no value is specified, all values are included in the selection criteria. Valid values are:

20 Send translate error

21 Send translated

29 Transaction detached -- send

30 Enveloped

31 Envelope error

41 Sent with errors

42 Send request error

43 Not sent network error

46 Send started status

48 Send requested

49 Sent to network

50 Accepted by network

51 Delivered by network

52 Purged by network

53 Recall requested

54 Recall request error

55 Recalled

61 Transaction accepted

62 Transaction rejected

63 Transaction accepted with errors

70 Received

71 Receive syntax error

72 Receive translated

73 Receive transaction error

74 Transaction detached–recv

For **TRANSFORM**, the default value is 21 if DIR(S) is specified. If DIR(R) is specified, the default value is 70.

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG
PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

TRXTIME

The time the transaction was added to the Document Store. This is also the time at which the transaction was translated to EDI standard format. The maximum length is eight.

This keyword is used with the following commands:

DOCUMENT DATA EXTRACT
ENVELOPE
ENVELOPE AND SEND
ENVELOPE DATA EXTRACT
HOLD
PRINT ACKNOWLEDGMENT IMAGE
PRINT ACTIVITY SUMMARY
PRINT EVENT LOG

PRINT STATUS SUMMARY
PRINT STATUS SUMMARY2
PRINT TRANSACTION DETAILS
PRINT TRANSACTION IMAGE
PURGE
QUERY
REENVELOPE
REENVELOPE AND SEND
RELEASE
REMOVE TRANSACTIONS
RETRANSLATE TO APPLICATION
TRANSACTION DATA EXTRACT
TRANSFORM
TRANSLATE TO APPLICATION
UNPURGE

USERID

The network user ID of the trading partner as entered in the `User ID` field of the trading partner profile. The maximum length is 32.

This keyword is used with the following commands:

NETWORK ACTIVITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT

USERPGM

Specifies the program that WebSphere Data Interchange links to just before writing a record during data extract processing. Your program should return a code to WebSphere Data Interchange indicating whether the record is written to the EDIQUERY file or discarded. If you do not specify this keyword, WebSphere Data Interchange writes the requested records to the EDIQUERY file. For more information, see Chapter 1, “Using The Utility,” on page 1 The maximum length is eight.

This keyword is used with the following commands:

ENVELOPE DATA EXTRACT
NETWORK ACTIVITY DATA EXTRACT
TRANSACTION DATA EXTRACT
TRADING PARTNER CAPABILITY DATA EXTRACT
TRADING PARTNER PROFILE DATA EXTRACT
TRANSACTION ACTIVITY DATA EXTRACT

VERIFY

Indicates whether the status of a transaction is verified before the transaction is put into an envelope. Valid values are:

Y Checks for correct status before an ENVELOPE or REENVELOPE operation.

(other)

Does not check status. This is the default.

This keyword is used with the following commands:

ENVELOPE
ENVELOPE AND SEND
REENVELOPE
REENVELOPE AND SEND

VERSION

The version specified on the send or receive map usage. The maximum length is 8. The default is blank.

This keyword is used with the DELETE USAGE command.

WRTCTLNO

Indicates whether trading partner control numbers are included when importing a trading partner profile member. Valid values are:

Y Imports control numbers with the trading partner. This is thedefault.

N Does not import control numbers. If this is a new trading partner, the control number is all zeros. If this trading partner already exists, the existing control numbers are not overwritten by the control numbers in the import file.

This keyword is used only with the IMPORT command.

XML

Indicates whether XML processing is required for the input data. Valid values are:

Y Requires XML processing.

N Does not require XML processing. This is the default.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
TRANSLATE AND ENVELOPE
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

XMLDICT

Identifies the PDS or HFS path for the XML dictionary files generated by the DTD conversion utility. This field is required for XML processing. The maximum length is 64.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
TRANSLATE AND ENVELOPE
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

XMLDTDS

Identifies the PDS or HFS path for the XML Schema or DTD members. This field is required for XML Schema and DTD processing. The maximum length is 64.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
TRANSFORM
TRANSLATE AND ENVELOPE
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

For more information about XML DTD resolution, see the *WebSphere Data Interchange for MultiPlatforms Programmer's Reference Guide*, SC34-6217-01.

XMLEBCDIC

Indicates whether the incoming XML data should be interpreted as EBCDIC data. Applies only to z/OS. Valid values are:

Y Interprets incoming XML data as EBCDIC data, regardless of the encoding on the XML declaration

N Uses the encoding on the XML declaration

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
TRANSLATE TO APPLICATION
TRANSFORM

XMLNS

Indicates if namespace processing for input XML data will be performed. If you are doing schema validation or your XML schema uses namespace qualified elements or attributes, you should set XMLNS to Y.

Y Do namespace processing for input XML data.

N Do not do namespace processing for input XML data. This is the default.

This keyword is used with the TRANSFORM command.

XMLSCHEMAVAL

Indicates if validation of the data against an XML schema is performed. The XML schema to validate should be in the directory specified by the XMLDTDS keyword.

Y Validate the data against an XML schema.

N Do not validate the data against an XML schema. This is the default.

A Validate the data against an XML schema only if one is specified. If no schema is specified, do not attempt to validate against a schema.

This keyword is used with the TRANSFORM command.

XMLSEGINP

Indicates whether record boundaries in the input file should be treated as line breaks. This keyword is ignored when XMLEBCDIC is set to N.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
TRANSLATE AND ENVELOPE
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD

XMLSPLIT

The XMLSPLIT keyword indicates whether the incoming XML documents should use the XML document split function when the corresponding XML Schema object or XML DTD object has elements assigned to the XML document split related fields. Valid values are:

Y XML documents will be split if the corresponding Schema object or DTD object specifies the document split information. This is the default.

N XML documents will not be split even if the corresponding Schema object or DTD object specifies the document split information.

This keyword is used with the TRANSFORM command.

XMLSTDID

For data transformation and XML source message processing. Overrides the XML DTD or Schema definition retrieval and processes the XML input message using default processing. The default is Y.

This keyword is used with the TRANSFORM command.

XMLVALIDATE

Indicates the level of validation applied to the incoming XML data. This keyword applies to XML documents described by DTDs. It does not apply to documents described by a schema. Valid values are:

0 Ignores external DTD references.

1 Uses the declared DTD for processing default attributes, entity references, and so on, but does not validate the data against the DTD. This is the default.

2 Fully validates the data against the DTD.

Note: DTD references will not be ignored if the XMLSCHEMAVAL keyword is specified with a value of Y or A.

This keyword is used with the following commands:

DEENVELOPE
DEENVELOPE AND TRANSLATE
TRANSLATE TO APPLICATION
TRANSFORM

XRECS

For data transformation and application data C and D data format as the target message. Indicates that XRECS should not be created in the output file for records exceeding 32000 bytes in length. The default is Y.

This keyword is used with the TRANSFORM command.

Chapter 4. Export/Import utility function

WebSphere Data Interchange provides an Export/Import utility function for updating databases in a batch environment. The Export/Import batch function uses three files:

1. Batch Control File (CTLFILE) containing the control information that describes the data being exported or imported.
2. Export/Import Files (E/I File) containing the data that is being imported or exported (in tagged or fixed format).
3. Print File (PRTFILE) containing a report on Export/Import activity.

The Export/Import utility enables you to choose one, any combination of, or all the following categories of WebSphere Data Interchange data for extracting (exporting) or loading (importing) EDI data:

1. EDI standards/standard transactions
2. Data formats
3. Maps
4. Control strings
5. Profiles
6. Tables

Each category is a complete set with or without the associated objects, depending on your choice. For example, for export/import of an EDI standard, the complete set includes:

- Standard definition
- Transaction definitions
- Transaction details
- Segment definitions
- Segment details
- Data element definitions

The associated objects are:

- Envelope standards
- Validation tables (if validation is required)

The categories and their associated objects are described on the following pages.

Sample programs included in the WebSphere Data Interchange product enable you to convert a fixed format (flat) file into a tagged import file for importing trading partner profile (TPPROF) members, send transaction usages/rules, and receive transaction usages/rules.

These sample programs and JCL include extensive documentation that describes how they work and how they can be used. These programs are:

- A COBOL program named EDIXF2T that is the main program executed.
- An Assembler program named EDIXTAGF, that is link edited with EDIXTAGF, and is a formatting service to create the tags.

- Sample JCL named EDIXF2T used to define the fixed flat file and to execute the conversion program EDIXF2T.

These programs and JCL are provided as source and can be modified, compiled, and link edited to suit your individual needs.

Export/Import control file (CTLFILE)

The export/import control file describes what data is being exported or imported. The control file can process multiple export or import requests.

For example, to export seven different data formats would require seven records in the control file.

The control file is specified with the `CTLFILE` keyword and is used with EXPORT and IMPORT commands. The suggested record format can be fixed or variable, but the record length should not exceed 120 bytes. You can use the `CTLTYPE` keyword to specify the type of control file. The following table describes the labels in the export/import control file.

Table 6. Export/Import control file labels

Label	Position	Length	Type	Description
CATEGORY	1	1	Char	Transaction category
REPLACE	2	1	Char	Replace named object
KEYID	3–32	30	Char	Object ID
ASSOBJ	33–54	22	Char	Associated objects
USAGETID	55–70	16	Char	Map name for usage import
MBRNAME	85–114	30	Char	Member name for profile import, or data format name for data format import
reserved	115–120	6	Char	Reserved for future use

Export/Import control file label descriptions

CATEGORY

Specifies the category of the transaction and is required for both export and import. Valid values are:

1 EDI standards/transaction sets
2 Data format dictionaries/data formats
3 Maps
4 Control strings
7 Profiles
8 Tables
9 All categories (import only)
A XML dictionaries/DTDs
B Global variables

D Server command
S Security segment

REPLACE

For import only. Indicates whether to replace the same named object in the database. This field does not apply to usages/rules or to profile members. It does apply to the parent categories of entire maps and profiles. This field applies only to the primary object identified by the category code. All associated objects that are imported overwrite duplicate entries currently in WebSphere Data Interchange without issuing a warning message. Valid values are:

1 Replaces the named object

(other) Does not replace the named object (default)

KEYID

Specifies the ID of the object to be exported or imported. The value must be left-justified. For exporting EDI standard transactions, the standard dictionary name is left-justified in the first 8 bytes. For importing specific EDI standard transaction sets, the import file must contain an EDI complete standard with all transaction sets included. If no key value is present for data format dictionaries, maps, or standards, all records are exported for the specified category.

This keyword is required for importing usages/rules. For importing other items, if you do not specify this keyword, all records for the category (specified in position 1) are processed.

This field specifies the profile type when performing a profile handline profiles. The profile type is one of the values listed under “Additional profile layouts” on page 269. Use the MBRNAME field of the control record to limit action to a specific profile entry. KEYID values are:

REQPROF
Mailbox Profile
SECUPROF
Network Security Profile
NETPROF
Network Profile
NETOP
Network Commands
APPDEFS
Application Default Profile
ADAMCTL
User Exits
LANGPROF
Language Profile
CONTRECV
Continuous Receive Profile
WebSphere MQ
WebSphere MQ Queue Profile

TPPROF
Trading Partner Profile
SYSPROF
CICS Performance
SERVICE
Service Profiles
MCD Message Content Descriptor profiles
EVENTDST
Event destination profile
DATAREF
Data reference profile
E Edifact Envelope Profiles
X X12 Envelope profiles
U UCS Envelope Profiles
I ICS Envelope profiles
T UNTDI/Tradacoms Envelope Profiles

ASSOBJ

For export only. Identifies an array specifying the associated objects. The array must contain either **1** (Yes) or any other value (No) for each associated object in the following order:

For EDI standards:
1. Validation tables
2. Envelope profiles
3. Envelope standards

For maps and control strings:
1. Usages/Rules
2. Control strings
3. Standard transactions (for DT maps source document definition)
4. Data formats (for DT maps target document definition)
5. Validation tables
6. Translation tables
7. User exit routines
8. Trading partner profiles
9. Translation exit routines
10. Network security profiles
11. Network profiles
12. Network command profiles
13. Envelope profiles
14. Envelope standards
15. Maps
16. Conversion of release 3.1 objects
17. Global variables
18. Validation maps
19. FA maps
20. Embedded maps

Note: There are no associated objects for XML DTDs or mapping global variables.

To export a usage/rule without its map, specify **1** for usages/rules (item 1) and **0** for maps (item 15).

To export validation maps and functional acknowledgment maps, specify **1** for usage/rules (item 1) and **1** for validation maps (item 18) and **1** for functional acknowledgment maps (item 19).

For import only. The array must contain a **1** (Yes) and any other value (No) for each associated object in the following order:

For maps:

1. Usages/Rules only (the KEYID value must not be blank.)
16. Conversion of release 3.1 objects

For data formats:

16. Conversion of release 3.1 objects

For EDI standards:

16. Conversion of release 3.1 objects

For Trading Partner Profiles, to export Send usages, Receive usages, and Rules. Send usages and Receive usages are identified by the TP Nickname in the member field of the export control record. All rows with a TPNICKNM or APPLTPID are selected and formatted as 7T5 and 7T6 export records. Rules are selected by the RECVTPID column equal to the TP nickname specified in the export control records. These are formatted as 7T7 export records:

1. Usages/Rules

All usages and rules with a Sending or Receiving trading partner equal to MBRNAME field specified in the export control record are exported.

Note: If MBRNAME field of the export control record is blank, all Trading Partner Profiles and associated Usages or Rules are exported.

USAGETID

For import only. Specifies the map name under which to import the usages/rules. If blank, the imported map name is used.

MBRNAME

Specifies the profile member name to export or import, or a transaction ID when exporting a specific EDI standard transaction. For data format dictionaries, specifies the dictionary name to be used. If the MBRNAME field of the export control record is blank, and TPPROF is used as the KEYID then all Trading Partner Profiles are exported.

Export/Import files

The export/import files are used for output when you export WebSphere Data Interchange data, or are used for input when you import WebSphere Data Interchange data. Export/Import files are sequential, contain variable length records, and must be allocated with a record format of V, LRECL=8152, and BLKSIZE=8156. There are several export/import files, each associated with the category of data they hold. The logical names are listed in Table 7.

Table 7. Export/Import file logical names

logical name	Category
EDIEISTD	EDI standards and standard transactions
EDIEIADF	Data formats
EDIEITBL	Tables
EDIEITPT	Maps
EDIEICST	Control strings
EDIEIPRF	Profiles

You can maintain export files and the data in those files by using the following z/OS/TSO ISPF utility functions:

DATASET

Allocate, rename, delete, and display export data set information

COPY

Copy export data set

DSLIST

Print, rename, delete, browse, and display export data set information

Note: In CICS, if you want to maintain the export files, define them as QSAM extrapartitioned TD queues at WebSphere Data Interchange system generation. To perform maintenance, close the queues in CICS, and use the ISPF utility functions in the z/OS/TSO environment. Then run batch JCL to execute the utility functions in the z/OS/TSO environment. CICS has the export/import files allocated and the data set names cannot be shared with ISPF utilities. When export queue maintenance is complete, open the queues in CICS. If the logical name associated with the queue specifies `DISP=`**OLD**, the queue is cleared when opened. If the logical name associated with the queue specifies `DISP=`**MOD**, the queue is not cleared when opened, and new export data is appended to the end of the queue.

WebSphere Data Interchange requires that the records contained in each Export/Import file follow a certain sequence:

- The first record of the Export/Import file must be the Common Control Record **(0C1** or **0C2)**. This record contains information such as the date, time and system release level.

- The last record of each group must be the Common End of Group Record **(000)**.
- The data records are placed between the **0C1**/**0C2** record and the first **000** record, and also between each **000** record and any subsequent **000** records. These records vary in purpose and length, and they must be written in the format described in Table 8 and Table 9 on page 220.

Figure 14 shows the set of records written to the EDIEIPRF logical name by this export.

```
Export/Import common control record              0C1
First trading partner profile header record      7P1
First trading partner profile detail record      7P2
Second trading partner profile detail record     7P2
Third trading partner profile detail record      7P2
Fourth trading partner profile detail record     7P2
Fifth trading partner profile detail record      7P2
Sixth trading partner profile detail record      7P2
Last trading partner profile detail record       7P2
Export/Import common end of group record         000
```

Figure 14. Example of records written to EDIEIPRF by export

The user exported seven trading partner profiles in tagged format. This resulted in WebSphere Data Interchange writing out the Common Control Record (0C1), and a Trading Partner Profile Header Record (7P1) that specifies that the following records are part of a trading partner profile export. This is followed by seven Trading Partner Profile Detail Records (7P2), one for each trading partner. The file ends with a Common End of Group Record (000).

Export/Import common control record (0C1/0C2)

The Common Control Record defines the format of the records (tagged or fixed), the user ID, date, time, and WebSphere Data Interchange version/release number. This record is required and it must be the first record in the Export/Import record data set.

Note: The first 3 bytes of this record are **0C1** or **0C2**, starting in column 1. The 0C1/0C2 record occurs only once in an export/import data set and is always the first record.

Table 8. Export/Import Common Control Record (0C1/0C2) fields

Field Name	Position	Length	Type	Field Description
Category	1	1	Char	Export/Import record category = **0**
Rectype	2–3	2	Char	Export/Import record type code Record type = **C1** (tagged format) Record type = **C2** (fixed format)
User ID	4–11	8	Char	User ID of person creating the export/import record data set
Date	12–17	6	Char	Creation date of export/import data set (YYMMDD)

Table 8. Export/Import Common Control Record (0C1/0C2) fields (continued)

Field Name	Position	Length	Type	Field Description
Time	18–23	6	Char	Creation time of export/import data set (HHMMSS)
Language	24–26	3	Char	WebSphere Data Interchange language English (United States) = **ENU**
DIVersRel	27–38	12	Char	WebSphere Data Interchange version and release: Vers 3 Rel 1 = 010301000000 Vers 3 Rel 2.1 = 010405000000 Vers 3 Rel 3 = 040600000000 Vers 4 Rel 1 = 010401000000

The following example of a completed 0C1 record depicts the user ID as SMITH, a creation date of **010419**, a creation time of **161616**, English as the language, and Version 4 Release 1 (**010401000000**) as the WebSphere Data Interchange version and release.

```
0C1 SMITH   010419161616ENU010401000000
```

Export/Import common end of group record (000)

The Common End of Group Record indicates the end of an export/import group of records. This record is required and must be the last record of an export/import group. Table 9 describes the fields of the common end of group record.

Table 9. Export/Import Common End of Group Record (000) Fields

Field Name	Position	Length	Type	Field Description
Category	1	1	Char	Export/Import record category = **0**
Rectype	2–3	2	Char	Export/Import record type = **00**

The Common End of Group Record contains only **000**, starting in column 1.

Export/Import file data area

The content data area varies depending upon which category of data you are importing or exporting. The following categories are supported:

- EDI standards and standard transactions
- Data formats
- Maps
- Control strings
- Profiles
- Tables

Regardless of the category chosen, the data portion (tagged or fixed) of an export/import file starts in column 4 of the record. Columns 1–3 are reserved for the Export/Import record ID (RECID). The Common Control Record (0C1/0C2) and Common End of Group Record (000) are fixed records. All records contained in the export/import

file data area might be tagged or fixed format records, except for the control string (G1) and document layout (L1), which are native database format.

Note: When fixed format record layouts are changed due to design changes in WebSphere Data Interchange , new fields are added to the end of the records.

The following tables define the order in which the fields are exported (position of fixed format fields), the length of fixed format fields (maximum length for tagged fields), and the type of data. Data with type **Char** can have any valid characters. Data with type **DEC** must have only decimal characters (0–9). When exported, data with type **HEX** is expanded to two hex representation characters (0–9, A-F) for each hex byte.

Export/Import trading partner profiles

The trading partner profile contains information about the companies that exchange business transactions with you. There must be one record for each trading partner that you wish to define.

The Export/Import format for defining trading partners includes the record types defined in Table 10.

Table 10. Trading partner record types

Record type	Description
7P1	One trading partner profile header record
7P2	One or more trading partner profile records
7P3	Zero or more trading partner contact records
7P4	One or more trading partner control number records
7P6	One or more message ID records
7P7	One or more destination properties records
7Z1	Zero or more contact records
7A1	At most, one trading partner comment record

The trading partner profile record group must be followed by the Export/Import Common End of Group Record (000). Figure 15 on page 222 illustrates a valid sequence of records written to logical name EDIEIPRF when exporting multiple trading partner profiles.

```
0C1 Export/Import common control record
7P1 Trading partner header record
7P2 Trading partner profile record
7P4 Trading partner control number record
7A1 Trading partner comment record
7P2 Trading partner profile record
7Z1 Contact record
7Z1 Contact record
7P3 Trading partner contact record
7P3 Trading partner contact record
7P3 Trading partner contact record
7P4 Trading partner control number record
7A1 Trading partner comment record
7P2 Trading partner profile record
7Z1 Contact record
7P3 Trading partner contact record
7P4 Trading partner control number record
7P2 Trading partner profile record
7P4 Trading partner control number record
7P2 Trading partner profile record
7P4 Trading partner control number record
000 Export/Import common end of group record
```

Figure 15. Example of multiple trading partner profiles

Importing new definitions to WebSphere Data Interchange

After the Export/Import file data set has been updated, you can submit the updates using the batch utility or with the WebSphere Data Interchange Import function.

Export/Import the trading partner profile header record (7P1)

The trading partner profile header record is required and must be the first record of a trading partner profile record group. A trading partner profile group is defined as one trading partner profile header record (7P1) followed by one or more trading partner profile records (7P2).

The trading partner profile header record contains the trading partner profile indicator and profile description. It is described in Table 11.

Table 11. Trading partner profile header record 7P1

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID =**TPPROF**
PROFDESC	12–46	35	Char	Profile/table description

Export/Import trading partner profile record (7P2)

The trading partner profile record defines the descriptive and control data that identify an individual trading partner and its processing options. Multiple 7P2 records can be created (one record per trading partner) to support the creation of multiple trading partner profiles.

For fixed format records, the fields must be in the position specified and for the length specified in the table.

For tagged files, the tags for the trading partner profile record (7P2) can be placed in any column or any order you choose. When importing data, all incoming variables must be delimited by parentheses. When exporting, outgoing variable data is placed between parentheses. Profile member tags and their associated maximum record lengths are listed in Table 12.

Table 12. Trading partner profile members

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID
TPNICKN	12–27	16	Char	Trading partner nickname
NETID	28–35	8	Char	Network ID
SYSQUAL	36	1	Char	System qualifier
SYSID	37–44	8	Char	System ID
ACCTID	45–76	32	Char	Account number
USERID	77–108	32	Char	User ID
INTQUAL	109–112	4	Char	Interchange ID qualifier
INTID	113–147	35	Char	Interchange ID
CONAME	148–187	40	Char	Company name
ADDR1	188–227	40	Char	Company address line 1
ADDR2	228–267	40	Char	Company address line 2
PHONE	268–292	25	Char	Contact phone
CONTACT	293–322	30	Char	Contact name
SNDPASS	323–336	14	Char	Interchange send password
RCVPASS	337–350	14	Char	Interchange receive password
SECPROF	351–358	8	Char	Network security profile ID
NETCLS	359	1	Char	Network message class
NETCHG	360	1	Char	Network charges code
NETACK	361	1	Char	Network acknowledgment
NETVCHK	362	1	Char	Destination verification code
NETRETN	363	3	Char	Retention period
NETEDIO	366	1	Char	EDI receive option
NETEDIP	367	1	Char	EDI processing override
STGFMTOV	368	1	Char	Storage format override
MACHTYPE	369	1	Char	Machine type
STGFMT	370	1	Char	Storage format
EOTID	371	1	Char	End of text/message delimiter
LOGENV	372	1	Char	Log standard data
FNCGRP	373	1	Char	Functional group code

Table 12. Trading partner profile members (continued)

Tag	Position	Length	Type	Description
SEDLM	374–375	2	Char	Subelement delimiter
DEDLM	376–377	2	Char	Data element delimiter
SEGDLM	378–379	2	Char	Segment delimiter
SEGSEP	380–381	2	Char	Segment ID separator
DECNOT	382	1	Char	Decimal notation
RLSCHAR	383–384	2	Char	Release character
INTCTLNO	385–393	9	Char	Interchange mask
GRPCTLNO	394–402	9	Char	Group mask
TRXCTLNO	403–411	9	Char	Transaction mask
COMMENT1	412–451	40	Char	Comment line 1
COMMENT2	452–491	40	Char	Comment line 2
NETCMDS	492–499	8	Char	Network commands file
TPDATALINE	500–531	32	Char	Trading partner data phone number
TIMEOUT	532–535	4	Dec	Data line communication timeout
SEGMENTED	536	1	Char	Segmented output option
SUFFIX	537–538	2	Char	File suffix
TPENVSUF	539–540	2	Char	Envelope profile suffix
TPGENRCV	541	1	Char	Enable generic receive
TPCMPRES	542	1	Char	Compression
TPSUPAD3	543–582	40	Char	Address line 3
TPSUPCTY	583–612	30	Char	City name
TPSUPST	613–614	2	Char	State code
TPSUPPST	615–629	15	Char	Postal code
TPSUPCON	630–659	30	Char	Country
TPSUPFAX	660–684	25	Char	Fax number
TPSUPU3	685–724	40	Char	Comment line 3
TPSUPU4	725–764	40	Char	Comment line 4
TPSUPU5	765–804	40	Char	Comment line 5
TPSUPU6	805–844	40	Char	Comment line 6
TPSUPU7	845–884	40	Char	Comment line 7
TPSUPU8	885–924	40	Char	Comment line 8
TPSUPU9	925–964	40	Char	Comment line 9
TPSUPU10	965–1004	40	Char	Comment line 10
DESCRIPT	1005–1034	30	Char	Description
TPTYPE	1035	1	Char	Trading partner type
PRIORITY	1036	1	Char	GXS Expedite priority

Table 12. Trading partner profile members (continued)

Tag	Position	Length	Type	Description
PROCESS	1037–1076	40	Char	Process ID
DESEP	1077–1078	2	Char	Repeating data element separator

The following example shows part of an export/import file that contains records for exporting/importing three trading partner profiles.

```
0C1SMITH   0140419161616ENU010103000000

7P1PROFID(TPPROF) PROFDESC(Trading partner profile)
7P2PROFID(TPPROF) TPNICKN(IBM)  NETID(IINR4)  SYSQUAL(4) NETACK(N)PASSWORD(N WPASS)
7P4TPNICKN(IBM) APPRECID(JONES) APPRECQ(01) INTCTLNO(000010000)
   GRPCTLNO(000015000) TRXCTLNO(000020000)
7P2PROFID(TPPROF) TPNICKN(TPT40) NETACK(Y)  SYSQUAL(4) NETID(IINR5) PHONE(555-1234)
7P2PROFID(TPPROF) TPNICKN(VAN04PR) CONTACT(John Doe) SNDPASS(Y) RCVPASS(Y)
7P4TPNICKN(VAN04PR) APPRECID(SMITH) APPRECQ(02) APPDOCID(850) INTCTLNO(000030001)
GRPCTLNO(000040000) TRXCTLNO(000045000)
000
```

Figure 16. Example of exporting trading partner profiles

Figure 16 shows how the 0C1 record is the first record in the Export/Import file, followed by profile data records with a 000 record as the last record. The 7P1 and 7P2 fields are the record IDs (RECIDs). They must be the first 3 bytes of the data record. In this example, the **7** indicates the category code, the **P** indicates the object being exported/imported (in this case, Profile), and the **1**, **2**, and **4** are internal WebSphere Data Interchange indicators.

Trading partner profile member field descriptions

Category code:

Must contain **7**.

Record type: Must contain **P2**.

Profile ID (PROFID): Must contain **TPPROF**.

Trading partner nickname (TPNICKN): The name you use to refer to this trading partner. Use the same name throughout WebSphere Data Interchange to refer to this trading partner by nickname.

Network ID (NETID): The name that identifies the network used to communicate with this trading partner. It must match the name of a member in the network profile.

System qualifier (SYSQUAL): For AT&T Global Network users, enter an **I** if intersystem addressing is required for this trading partner (Network reference: DTBLTYP). Enter the ID of the other system in the next field.

System ID (SYSID): For intersystem addressing, the ID of the system responsible for the receiver's account. For AT&T Global Network users, the ID is limited to three characters (Network reference: DTBLID).

Account number (ACCTID): The account number assigned to your trading partner by the network. The entry must be left-justified. For sending and receiving EDI in ISA/IEA envelopes, the last position must be blank. The combination of this field and the User ID field must be unique for all members in the profile.

If the interchange ID is blank, the account number and user ID make up the receiver ID in the interchange envelope. UCS (BG/EG) envelopes are an exception, where the phone number field contains the receiver ID.

User ID (USERID): The user ID assigned to your trading partner by the network. The entry must be left-justified. The combination of this field and the account number field must be unique for all members in the profile.

If the interchange ID field is blank, the account number and user ID make up the receiver ID in the interchange envelope. UCS (BG/EG) envelopes are an exception, where the phone number field contains the receiver ID.

Interchange ID qualifier (INTQUAL): The type of interchange ID entered in the interchange ID field such as a Dun and Bradstreet number (DUNS). The EDI standard defines these codes. If the interchange ID is blank, the enveloper takes the qualifier from the envelope profile member. The combination of this field and the interchange ID field must be unique for each member unless the fields are left blank.

Interchange ID (INTID): The ID used as the interchange receiver ID (recipient) when you send to this partner and as the interchange sender ID when you receive from this partner. If this field is blank, the enveloper uses the account number and user ID (or phone number for BG/EG interchanges). The combination of this field and the interchange ID qualifier field must be unique for each member unless the fields are left blank.

Company name (CONAME): The name of the trading partner's company.

Company address line 1 (ADDR1): The first line of the trading partner's address.

Company address line 2 (ADDR2): The second line of the trading partner's address.

Contact phone (PHONE): The trading partner's phone number (free-form). For UCS (BG/EG) enveloping, if the interchange ID is blank, the enveloper uses the phone number as the interchange ID.

Contact name (CONTACT): The name of the person you communicate with when dealing with this trading partner.

Interchange send password (SNDPASS): The password agreed upon by you and your trading partner for sending to this trading partner. This value maps to the `password data type` in the interchange envelope.

Interchange receive password (RCVPASS): The password agreed upon by you and your trading partner for receiving from this trading partner. If this value matches the interchange password (password data type) that was received, translation occurs.

Network security profile ID (SECPROF): The ID of the default network security profile member that specifies the encryption and authentication processes that apply to EDI data for this trading partner. This profile member is always used when receiving from this partner. It is used by default when sending to this partner unless the send usage/rule for the map specifies a different member in the `Group security profile name` field or the `Trans security profile name` field.

Network message class (NETCLS): Applies only to sending. Indicates any special status (for example, test) of the data being sent. For AT&T Global Network users (Network reference: MSGNCLS), **T** indicates test status, and a blank indicates normal status.

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

Network charges code (NETCHG): Indicates how charges are shared between sender and receiver. For AT&T Global Network users (Network reference: MSGCHRG), valid values are:

1 Receiver pays all charges.

2 Receiver pays all charges if agreed to; otherwise, charges are split between sender and receiver.

3 Receiver pays all charges if agreed to; if receiver does not agree to pay all charges, charges are split between sender and receiver if agreed to; otherwise, the sender pays all charges.

4 Charges are split between sender and receiver if agreed to; otherwise, the sender pays all charges.

5 Charges are split between sender and receiver.

6 Sender pays all charges.

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

Network acknowledgment (NETACK): Indicates which network acknowledgments (receipt, delivery, purge) you want to receive when sending to this partner. For AT&T Global Network users (Network reference: MSGRCPT), valid values are:

(blank) No acknowledgments

R Receipt only

D Delivery only

B Both receipt and delivery

A Purge only

C Both receipt and purge

E Either purge or delivery

F Receipt and either delivery or purge

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

Destination verification code (NETVCHK): Indicates whether the destination is verified before the data is sent. For AT&T Global Network users (Network reference: MSGVCHK), valid values are:

N Does not request verification (default)

Y Requires verification

F Requests verification, but sends even if the destination is not verified (useful for intersystem addressing)

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

Retention period (NETRETN): The number of days that the data is kept in the network mailbox before being purged, if it is not received.

If you are using more than one network, contact a representative from the network you are using or see the appropriate manual to determine the correct values to use for this field. The following pertains to the AT&T Global Network and to GXS Expedite Base 4.2 and higher.

Valid values are **0** through **180**. For installations in the U.S., the default retention period is **30** days. For installations outside the U.S., contact your marketing representative for your default value.

If you specify **0** or blank, Information Exchange retains the data for the default period. In addition, if you specify a value larger than the maximum for your system, Information Exchange retains the data for the default period.

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

Note: The size of this field for GXS Expedite/CICS is two digits, meaning the valid values for GXS Expedite/CICS are **0–99**. The two leftmost digits entered in this field are used by GXS Expedite/CICS as the retention period. For example: if you enter 180, the retention period value sent to GXS Expedite/CICS is **18**.

EDI receive option (NETEDIO): Indicates whether you want EDI segments stored in the file as separate records. For AT&T Global Network users (Network reference: EDIOPT), valid values are:

Y Ends records at the segment delimiter (default)

N Does not end records at the segment delimiter

EDI processing override (NETEDIP): Indicates whether you want the EDI data you receive to have special EDI processing, which breaks records by the segment delimiter. For AT&T Global Network users (Network reference: EDIPROC), valid values are:

Y Performs EDI processing if the common data header indicates that the data is in EDI standard format (default)

N Does not perform EDI processing, regardless of the values in the common data header

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

Storage format override (STGFMTOV): If you are using more than one network, contact a representative from the network you are using to determine the available options.

Information Exchange uses common data header control information that is used to send or receive information electronically. The control header contains information such as the type of record, original record format, sending system type, type of data being sent, whether the data is in an EDI format, and a unique record number for tracking.

If you are using IBM GXS Expedite Base/MVS (IEBASE), see the `DLMOVERRIDE` command option keyword for an up-to-date list of acceptable values. Valid values are:

Y Formats the data according to the `DELIMITED` parameter, even if the common data header (CDH) indicates a delimiter type

N Ignores the value in the `DELIMITED` parameter, and formats the data according to the record type (if any) in the CDH (default)

If there is no common data header, the format indicated in the `Storage format` field is used. If your request does not specify a trading partner, the information is taken from the mailbox (requestor) profile.

Machine type (MACHTYPE): This field is not currently supported.

Storage format (STGFMT): Indicates to the network how data is stored for free-form messages and files. When determining what option to select, you should consider the type of data you want to send and how the file is received. If you are using more than one network, contact a representative from the network you are using to determine the available options. Valid values are:

C Stores each record with a carriage return and line-feed character (CRLF) and uses the end of file (EOF) character to mark the end of file. These characters are represented and stored as hex values 0D0A and 1A respectively. This option is generally used to send files containing program source code with variable length records. Output records include data up to the carriage return and line feed characters (CRLF).

L Precedes each record with a 2-byte hexadecimal record length. Select this

option for sending data in fixed format or for sending binary data. The output record length is determined by the value in the 2 bytes containing the record length.

N Stores the data as it is received (default). Output records are built based on the record length of the data set allocated to receive the data.

If you are using GXS Expedite/CICS, see the DTYPE field for an up-to-date list of valid values:

A Stores each record with a carriage return and line-feed character (CRLF) and uses the end of file (EOF) character to mark the end of file. These characters are represented and stored as hex values 0D0A and 1A respectively. This option is generally used to send files containing program source code with variable length records. Output records include data up to the carriage return and line feed characters (CRLF).

B Precedes each record with a 2-byte hexadecimal record length. Select this option for sending data in fixed format or for sending binary data. The output record length is determined by the value in the 2 bytes containing the record length.

E Stores each record based on the delimiters found in the EDI standard data.

O Stores data as it is received (free-form).

If your request does not specify a trading partner, this information is taken from the mailbox (requestor) profile.

End of text or message delimiter (EOTID): The character that signifies to the network the end of the data. It applies to free-form messages and data files, not to EDI standard transactions. (Network reference: EORCHAR, EOMIND).

Log standard data (LOGENV): Indicates whether the translator should include EDI standard (untranslated) data when logging an image of a transaction. For EDI data with errors, the EDI standard data is logged even if you enter **N**.

This field can be overridden by the `Log standard data` field of the Application Defaults (APPDEFS) profile. Valid values are:

Y Includes EDI standard (untranslated) data when logging transaction image

N Does not include EDI standard (untranslated) data when logging transaction image

Functional group code (FNCGRP): Indicates whether the enveloper should create functional groups for transactions with type **E** envelopes. (Functional groups are always created for types **I**, **U**, and **X**; they are never created for type **T**). If you choose not to have functional groups, the different message types are enveloped together. Valid values are:

Y Creates functional groups

N Does not create functional groups

Subelement delimiter (SEDLM): The character (in hex notation) that separates subelements (also called component data elements) in a transaction set. An entry here (other than 00 or 40) overrides the character specified by the EDI standard.

If entered, this character must be different from the characters specified for the data element delimiter, segment delimiter, segment ID separator, decimal notation, and release character fields, with this exception: the subelement delimiter and the data element delimiter can be the same for envelopes with types **I**, **U**, and **X**.

Data element delimiter (DEDLM): The character (in hex notation) that separates the data elements in a transaction set. An entry here (other than 00 or 40) overrides the character specified by the EDI standard.

If entered, this character must be different from the characters specified for the subelement delimiter, segment delimiter, segment ID separator, decimal notation, and release character fields, with this exception: the subelement delimiter and the data element delimiter can be the same for envelopes with types **I**, **U**, and **X**.

Segment delimiter (SEGDLM): The character (in hex notation) that marks the end of each segment in a transaction set. An entry here (other than 00 or 40) overrides the character specified by the EDI standard.

If entered, this character must be different from the characters specified for the subelement delimiter, data element delimiter, segment ID separator, decimal notation, and release character fields.

Segment ID separator (SEGSEP): The character (in hex notation) that separates the segment ID and the first data element in a segment. An entry here (other than 00 or 40) overrides the character specified by the EDI standard.

If entered, this character must be different from the characters specified for the subelement delimiter, data element delimiter, segment delimiter, decimal notation, and release character fields.

Decimal notation (DECNOT): The character that represents the decimal point in a transaction set. For type **E** envelopes, an entry here overrides the character specified by the EDI standard. For all other types, a period represents the decimal point.

If entered, this character must be different from the characters specified for the subelement delimiter, data element delimiter, segment delimiter, segment ID separator, and release character fields.

Release character (RLSCHAR): For type **E** and **T** envelopes, the character (in hex notation) that is used to indicate where a delimiter is being used as part of the data. An entry here (other than 00 or 40) overrides the character specified by the EDI standard.

If entered, this character must be different from the characters specified for the subelement delimiter, data element delimiter, segment delimiter, segment ID separator, and decimal notation fields.

Interchange mask (INTCTLNO): The initial reference number that the enveloper places in the **CN** `data type` of the interchange header and trailer. This value is used as the base value for each trading partner/receiver ID combination. This field does not represent the current control number for this trading partner. Use the `Control Number` option on the Member List panel to request control number information.

Group mask (GRPCTLNO): The initial reference number or special codes that the enveloper places in the **CN** `data type` of the functional group header and trailer. This value is used as the base value for each trading partner/receiver ID combination. Use caution when updating this field. This field does not represent the current control number for this trading partner. Use the `Control Number` option on the Member List panel to request control number information.

Transaction mask (TRXCTLNO): The initial reference number or special codes that the enveloper places in the `CN data type` of the transaction set header and trailer. This value is used as the base value for each trading partner/receiver ID combination. Use caution when updating this field. This field does not represent the current control number for this trading partner. Use the `Control Number` option on the Member List panel to request control number information.

Comment line 1 (COMMENT1): Free-form notes about the trading partner.

Comment line 2 (COMMENT2): Free-form notes about the trading partner.

Network commands file (NETCMDS): The name of a PDS member to be allocated the z/OS logical name EDINTCMD. This member contains the commands that you want to pass to the network.

WebSphere Data Interchange reads the commands from the PDS member and writes the commands to the network input file specified in the network profile member after all substitutable variable tags are resolved by WebSphere Data Interchange .

TP data phone number (TPDATALINE): The phone number to dial to connect to your trading partner directly. This is the number used by your computer to talk directly to your trading partner's computer. The Contact phone field contains the voice phone number to call a human (not a computer) at your trading partner's site.

Data line communication timeout (TIMEOUT): Specifies the maximum time that the communications line can be idle without dropping the connection. If the data line is idle for a time greater than the value entered, the line is dropped. If you specify a trading partner when requesting network activity, the value for this field is taken from the trading partner profile. Otherwise, the value for this field is taken from the mailbox (requestor) profile.

Segmented output option (SEGMENTED): Indicates whether you want EDI segments stored in the output file as separate records. Valid values are:

Y Ends records at the segment delimiter

N Does not end records at the segment delimiter (default)

File suffix (SUFFIX): A 2-character suffix for the logical name used to store the results of a fixed-to-fixed translation. The logical name is taken from the `Application file name` field of the target data format.

This 2-character suffix is appended to the `Application file name` or, if the file name is longer than 6 characters, overwrites the last two characters of the `Application file name`. Use this suffix to separate the results of a fixed-to-fixed translation by trading partner. If you do not want to separate results by trading partner, either leave the suffix blank or assign all the logical names to the same physical file.

For CICS, you can use the suffix to identify a unique TS queue for each trading partner.

Envelope profile suffix (TPENVSUF): A 2-character suffix for a generic envelope profile member name. A generic envelope profile name has an ampersand (&) as the first character of the name followed by a 1 to 6-character common name.

This 2-character suffix is used to generate the envelope profile member name for enveloping transactions. Use this suffix to use different envelope profile members for different trading partners. To use the same envelope profile member for several trading partners, enter the same suffix in each of the trading partner profiles, or leave the suffix blank to use a common envelope profile member (with the same name as the common name in the usage/rule). You must define an envelope profile member for each combination of common name/suffix value specified in a trading partner profile.

The envelope profile member name is specified on the Send and Receive Trading Partner Usage panels. The generic form of the name is most useful for generic usages/rules but can be used for any usage/rule.

Enable generic receive (TPGENRCV): Indicates whether transactions received from this trading partner can be translated using generic receive trading partner usages/rules. Valid values are:

Y Translates using generic receive trading partner usages and rules

N Does not translate with generic usages/rules (default)

Compression (TPCMPRES): Indicates whether GXS Expedite Base/MVS or GXS Expedite/CICS should compress your data prior to transmission. Valid values are:

N Does not compress data (default).

Y Compresses data prior to transmission. If the file sent contains multiple EDI envelopes, all envelopes are sent compressed.

T Conditionally compresses data prior to transmission. For GXS Expedite Base/MVS, compression occurs if the `SENDER`, `RECEIVER`, and `COMPRESS` parameters are listed in the CPLOOKUP EXPFILE file. This file defines a series of paired receivers and senders, and for each pair, indicates whether compression should be performed. For GXS Expedite/CICS, compression occurs if the sender/receiver pair is found in the lookup table and `COMPRESS` equals **Y** in the lookup table.

If a trading partner profile member is specified on the send request, the compression value is taken from the trading partner profile. Otherwise, this information is taken from the mailbox (requestor) profile.

Address line 3 (TPSUPAD3): The third line of the trading partner's address.

City name (TPSUPCTY): The trading partner's city.

State code (TPSUPST): The trading partner's state.

Postal code (TPSUPPST): The trading partner's zip code.

Country (TPSUPCON): The trading partner's country.

Fax number (TPSUPFAX): The trading partner's fax number.

Comment line 3 (TPSUPU3): Free-form remarks.

Comment line 4 (TPSUPU4): Free-form remarks.

Comment line 5 (TPSUPU5): Free-form remarks.

Comment line 6 (TPSUPU6): Free-form remarks.

Comment line 7 (TPSUPU7): Free-form remarks.

Comment line 8 (TPSUPU8): Free-form remarks.

Comment line 9 (TPSUPU9): Free-form remarks.

Comment line 10 (TPSUPU10): Free-form remarks.

Description (DESCRIPT): A free-form description of the member.

Trading partner type (TPTYPE): Indicates the type of trading partner this definition represents. Valid values are:

E EDI trading partner (default). Considered to be external to your business organization. The interchange control numbers are generated using the EDI trading partner ID.

A Application trading partner. Considered to be internal to your business organization. For example, divisions or departments might be internal trading partners. Use this setting when no specific EDI trading partner is defined, or in combination with an EDI trading partner. Interchange control numbers are generated using the application and EDI trading partner combination. Use this trading partner type with centralized EDI when multiple application trading partners do business with the same EDI trading partner.

B Both EDI trading partner and application trading partner. Considered to be both external (EDI) and internal (application). Use this setting when your

organization provides EDI translation services to customers who are trading partners. The value in this field is directly related to the usage/rule setup.

For more information, see the *WebSphere Data Interchange for MultiPlatforms User's Guide*, SC34-6215-01.

Priority (PRIORITY): Indicates whether messages are sent through GXS Expedite Base/MVS or GXS Expedite/CICS with normal delivery or priority delivery. Valid values are:

(blank) Normal delivery

P High priority

For more information, see the SEND commands in the *GXS Expedite Base/MVS Programming Guide* or to Customizing and Developing Applications with GXS Expedite/CICS.

If a trading partner profile member is specified on the send request, this value is taken from the trading partner profile. Otherwise, this information is taken from the mailbox (requestor) profile.

Process: The process name used with data transformation to associate with this trading partner, such as `PRODUCTION_PURCHSG_V1R1`. You can create data transformation rules for a group of maps (creating a process) and then associate trading partners with the process. If you change the map rules for the process, all trading partners associated with the process use the changed maps automatically.

In the same manner, you can use the process name to create classes of trading partners such as production, non-production, financial institutions, XML trading partners, or EDI trading partners. You can create data transformation map rules to associate with each trading partner class.

Repeating data element separator (DESEP): The hex character separator placed between repeating data elements for all transactions sent to this trading partner. A value in this field (other than 00 or 40) overrides the separator character defined in the EDI standard. This character must be different from the characters specified for the subelement delimiter, data element delimiter, segment delimiter, segment ID separator, decimal notation, and release characters. This hex character should not be found within the EDI standard data.

Event destination profile (7P2)

This file contains destination information for Print Files, XML Print Files, and ADF Print Files generated during Utility invocations. The Event Destination profile:

Table 13. EVENTDST fields

Description	Position	Length
Profile ID (EVENTDST)	0–7	8
Destination ID (key)	8–23	16

Table 13. EVENTDST fields (continued)

Description	Position	Length
Description	24–53	30
Accgroup	54–83	30
Folder	84–147	64
Profile Active Flag	148	1
File Type To Route	149	1
Routing Severity Threshold	150–151	2
Application ID	152–159	8
Condition Code Override	160–163	4
XML Header Wanted?	164	1
Destination Type	165–166	2
Destination Name	167–174	8

Trading partner contact definition (7P3)

Trading partner contact information is exported or imported automatically when a trading partner record is exported or imported. Each trading partner contact record specifies a contact that is related to a trading partner. The fields are described in Table 14..

Table 14. Trading partner contacts definition

Tag	Position	Length	Type	Description
TPNICKN	4–19	16	Char	Trading partner nickname
TPCONNM	20–59	40	Char	Trading partner contact name

Trading partner control numbers (7P4)

Trading partner control numbers are exported or imported automatically when a trading partner is exported or imported. Each trading partner control number specifies a control number that is related to a trading partner. The fields are described in Table 15.

Table 15. Trading partner control numbers

Tag	Position	Length	Type	Description
TPNICKN	4–19	16	Char	Trading partner nickname
APPRECID	20–54	35	Char	Receiver ID
APPRECQ	55–58	4	Char	Receiver qualifier
APPDOCID	59–66	8	Char	Transaction ID, message ID, or document definition name
INTCTLNO	67–75	9	Char	Interchange control number
GRPCTLNO	76–84	9	Char	Group control number
TRXCTLNO	85–93	9	Char	Transaction control number

Message ID table (7P6)

This file contains information message generated during Utility invocations. The Event Destination message ID table:

Table 16. Message ID table fields

Description	Position	Length
Profile ID	0–7	8
Profile key	8–23	16
Message ID	24–29	6
Flags	30–69	40

Destination properties table (7P7)

This file contains information message generated during Utility invocations. The Event Destination properties table:

Table 17. Message ID table fields

Description	Position	Length
Profile ID	0–7	8
Destination ID	8–23	16
Property name	24–87	64
Property value	88–343	256

Comments definition (7A1)

Comments are exported or imported automatically when a trading partner is exported or imported. The comment record specifies textual data about a trading partner. The fields are described in Table 18.

Table 18. Comments definition

Tag	Position	Length	Type	Description
CMTTYPE	4–11	8	Char	Comments entity type
CMTKEY	12–46	35	Char	Comments entity type key
CMTTEXT	47–2094	2048	Char	Trading partner contact comments

Contact definition (7Z1)

Contact information is exported or imported automatically when a trading partner record is exported or imported. The contact record specifies data about a contact. The fields are described in Table 19 on page 238.

Table 19. Contacts definition

Tag	Position	Length	Type	Description
TPCONNM	4–43	40	Char	Trading partner contact name
TPCONTLE	44–83	40	Char	Trading partner contact job title
TPCONAD1	84–123	40	Char	Trading partner contact address line 1
TPCONAD2	124–163	40	Char	Trading partner contact address line 2
TPCONAD3	164–203	40	Char	Trading partner contact address line 3
TPCONCTY	204–233	30	Char	Trading partner contact city
TPCONST	234–235	2	Char	Trading partner contact state
TPCONPST	236–250	15	Char	Trading partner contact postal code
TPCONCON	251–280	30	Char	Trading partner contact country
TPCONPHN	281–305	25	Char	Trading partner contact phone
TPCONFAX	306–330	25	Char	Trading partner contact fax number
TPCONOTH	331–355	25	Char	Trading partner contact other phone
TPCONEML	356–395	40	Char	Trading partner contact e-mail address
CMTTEXT	396–2443	2048	Char	Trading partner contact comments

Export/Import EDI standard records

EDI standards can be exported/imported using tagged or fixed format files. The EDI standard/transaction set includes:

1Y1 Dictionary record

1Y2 Transaction header record

1Y3 Transaction detail record

1Y4 Segment header record

1Y5 Segment detail record

1Y6 Data element header record

1Y7 Data element detail record

1Y8 Segment notes record

1Y9 Transaction notes record

1YA Composite data element notes record

1YB Envelope detail record

Three types of associated objects can be exported using tags:
1. Validation tables
2. Envelope profile
3. Envelope standard

In Table 20, you can export all of the data elements in a particular segment. Since a segment is made up of many data element usages/rules and definitions, multiple Y6 and Y7 records are shown here.

Table 20. Exporting all data elements in a particular segment

Record Type	Record Name
0C1	Export/Import common control record
1Y1	EDI standard dictionary record
1Y2	EDI standard transaction header record
1Y3	EDI standard transaction detail record
1Y4	EDI standard segment header record
1Y5	EDI standard segment detail record
1Y6	EDI standard data element header record
1Y7	EDI standard data element detail record
1Y6	EDI standard data element header record
1Y7	EDI standard data element detail record
1Y8	EDI standard segment notes record
1Y9	EDI standard transaction notes record
1YA	EDI standard composite data element notes record
1YB	EDI standard envelope details record
000	Export/Import common end of group record

The following example shows part of an export/import file data set that contains records for exporting/importing one transaction of an EDI standard. The transaction has one segment with three data elements.

```
0C1ADMIN   011116151224ENU010401000000

1Y1 STDID(AARV4R4) AGENCY(X) VERSION(40) RELEASE(40) INDUSTRY(RAIL) DICTENVFLAG(D)
        INTENV(X) STDDESC(Rail Carrier 4040)  ALLTRXS(N)
1Y2 STDID(AARV4R4) STDTRID(426) FNGRPID(SR) STDDESC(Rail Revenue Waybill) STDPURP(Purpose)
1Y3 STDID(AARV4R4) STDTRID(426) TABLENO(1) POSNO(20) SEGID(ZR) SEGREQ(M) SEGMAX(1) UNLIMMAX(0)
UNLPREP(0) 1Y4 STDID(AARV4R4) SEGID(ZR) STDDESC(Waybill Reference Identification)
STDPURP(To transmit identity and reference information of the waybill)
1Y5 STDID(AARV4R4) SEGID(ZR) POSNO(1) DEID(762) DEREQ(M) 1Y6 STDID(AARV4R4) DEID(762)
DETYPE(ID) DEMIN(1) DEMAX(1) CODELIST(762R440) STDDESC(Waybill Response Code)
STDPURP(Code indicating a waybill response)
1Y9 STDID(AARV4R4) STDTRID(426) TABLENO(1) POSNO(241) NOTETYPE(N) NOTEPARA(1) RELATION(CM)
        STDNOTE(One R2B segment is needed for each line haul carrier)
1Y8 STDID(AARV4R4) SEGID(ZR) POSNO(4) NOTETYPE(N) NOTEPARA(1) RELATION(P) REL1(4) REL2(5)
000
```

EDI standard dictionary record (1Y1)

This record defines the EDI standard being used. The fields are described in Table 21 on page 240.

Table 21. EDI standard dictionary record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
AGENCY	34–41	8	Char	Agency issuing the standard
VERSION	42–49	8	Char	Version of the standard
RELEASE	50–57	8	Char	Release level of the standard
INDUSTRY	58–65	8	Char	Industry code of the standard
DICTENVFLAG	66	1	Char	Standard type flag (dictionary or envelope)
INTENV	67–96	30	Char	Interchange envelope name
STDDESC	97–176	80	Char	Description of the standard
STDPURP	177–688	512	Char	Purpose of the standard
CMTTEXT	689–2736	2048	Char	Comments
ALLTRXS	2737	1	Char	All transactions flag

EDI standard transaction header record (1Y2)

This record defines the transaction header fields being used. The fields are described in Table 22.

Table 22. EDI standard transaction header record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
STDTRID	34–41	8	Char	Transaction ID
FNGRPID	42–49	8	Char	Function group ID
STDDESC	50–129	80	Char	Description of the standard
STDPURP	130–2177	2048	Char	Purpose of the standard
CMTTEXT	2178–3977	1800	Char	Comments

EDI standard transaction detail record (1Y3)

This record defines the detail fields being used. If you are importing a transaction set, you can define multiple transaction detail records. The fields are described in Table 23.

Table 23. EDI standard transaction detail record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
STDTRID	34–41	8	Char	Transaction ID
TABLENO	42–47	6	Char	Table ID
POSNO	48–53	6	Char	Position number

Table 23. EDI standard transaction detail record (continued)

Tag	Position	Length	Type	Description
SEGID	54–61	8	Char	Segment ID
SEGREQ	62	1	Char	Segment required flag
SEGMAX	63–73	11	Char	Number of times segment can be repeated
UNLIMMAX	74–79	6	Char	No limit on the number of times segment can be repeated
NLOOPID	80–85	6	Char	Loop ID
LPREP	86–96	11	Char	Number of times loop can be repeated
UNLPREP	97–102	6	Char	No limit on the number of times loop can be repeated
LPLEV	103–108	6	Char	Loop level

EDI standard segment header record (1Y4)

This record defines the segment header fields being used. The fields are described in Table 24.

Table 24. EDI standard segment header record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
SEGID	34–41	8	Char	Segment ID
STDDESC	42–121	80	Char	Description of the standard
STDPURP	122–2169	2048	Char	Purpose of the standard
CMTTEXT	2170–3969	1800	Char	Comments

EDI standard segment detail record (1Y5)

This record defines the segment detail fields being used. The fields are described in Table 25.

Table 25. EDI standard segment detail record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
SEGID	34–41	8	Char	Segment ID
POSNO	42–47	6	Char	Position number
DEID	48–55	8	Char	Data element ID
DEREQ	56	1	Char	Data element required flag
SEGMAX	57–67	11	Char	Number of times segment can be repeated

Table 25. EDI standard segment detail record (continued)

Tag	Position	Length	Type	Description
HIPPAUSG	68	1	Char	HIPPA usage

EDI standard data element header record (1Y6)

This record defines the data element header fields being used. The fields are described in Table 26.

Table 26. EDI standard data element header record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
DEID	34–41	8	Char	Data element ID
DETYPE	42–43	2	Char	Type of data element
DEMIN	44–54	11	Char	Minimum length of data element
DEMAX	55–65	11	Char	Maximum length of data element
CODELIST	66–73	8	Char	Code list name
STDDESC	74–123	80	Char	Description of the standard
STDPURP	124–2171	2048	Char	Purpose of the standard
CMTTEXT	2172–3971	1800	Char	Comments

EDI standard data element detail record (1Y7)

This record defines the data element detail fields being used. The fields are described in Table 27.

Table 27. EDI standard data element detail record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
COMPID	34–41	8	Char	Composite data element ID
POSNO	42–47	6	Char	Position number
DEID	48–55	8	Char	Data element ID
DEREQ	56	1	Char	Data element required flag
SEGMAX	57–67	11	Char	Number of times segment can be repeated

EDI standard segment notes record (1Y8)

This record defines the segment note fields being used. The fields are described in Table 28 on page 243.

Table 28. EDI standard segment notes record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
SEGID	34–41	8	Char	Segment ID
POSNO	42–47	6	Char	Position number
NOTETYPE	48	1	Char	Type of note
NOTEPARA	49–54	6	Char	Paragraph number
RELATION	55–56	2	Char	Relationship of one data element to another
REL1	57–62	6	Char	Related data element reference number
REL2	63–68	6	Char	Related data element reference number
REL3	69–74	6	Char	Related data element reference number
REL4	75–80	6	Char	Related data element reference number
REL5	81–86	6	Char	Related data element reference number
REL6	87–92	6	Char	Related data element reference number
REL7	92–98	6	Char	Related data element reference number
REL8	99–104	6	Char	Related data element reference number
REL9	105–110	6	Char	Related data element reference number
REL10	111–116	6	Char	Related data element reference number
STDNOTE	117–1140	1024	Char	Text of the note

EDI standard transaction notes record (1Y9)

This record defines the transaction note fields being used. The fields are described in Table 29.

Table 29. EDI standard transaction notes record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
STDTRID	34–41	8	Char	Transaction ID
TABLENO	42–47	6	Char	Table number
POSNO	48–53	6	Char	Position number
NOTETYPE	54	1	Char	Type of note
NOTEPARA	55–60	6	Char	Paragraph number
RELATION	61–62	2	Char	Relationship of one data element to another
REL1	63–68	6	Char	Related data element reference number
REL2	69–74	6	Char	Related data element reference number
REL3	75–80	6	Char	Related data element reference number
REL4	81–86	6	Char	Related data element reference number

Table 29. EDI standard transaction notes record (continued)

Tag	Position	Length	Type	Description
REL5	87–92	6	Char	Related data element reference number
STDNOTE	93–1116	1024	Char	Text of the note

EDI standard composite data element notes record (1YA)

This record defines the composite data element note fields being used. The fields are described in Table 30.

Table 30. EDI standard composite data element notes record

Tag	Position	Length	Type	Description
STDID	4–33	30	Char	EDI standard dictionary name
COMPID	34–41	8	Char	Composite data element ID
POSNO	42–47	6	Char	Position number
NOTETYPE	48	1	Char	Type of note
NOTEPARA	49–54	6	Char	Paragraph number
RELATION	55–56	2	Char	Relationship of one data element to another
REL1	57–62	6	Char	Related data element reference number
REL2	63–68	6	Char	Related data element reference number
REL3	69–74	6	Char	Related data element reference number
REL4	75–80	6	Char	Related data element reference number
REL5	81–86	6	Char	Related data element reference number
STDNOTE	87–1110	1024	Char	Text of the note

Exporting and importing data formats

Data formats can be exported/imported using either tagged or fixed format files. The data format set includes:

2W1 Data format dictionary

2W2 Data format record ID information

2W3 Data format leader record

2W4 Data format loop record

2W5 Data format record record

2W6 Data format structure record

2W7 Data format field record

2W8 Data format header details record

2W9 Data format loop details record

2WA Data format record details record

2WB Data format structure details record

There are no associated objects for data formats.

The following example shows part of an export/import file data set that contains records for exporting/importing one data format.

```
0C1DD5TST1 011025134023ENU010401000000

2W1  DICTIONARYNAME(MMTHL1_DICTIONARY)  DESC(MAUI MAP TEST - HL1 - SUW)  ALLADFS(N)

2W2  RECORDIDINFONAME(MMTHL1_RECORDID)  DESC(MAUI MAP TEST - HL1 - SUW)  ADFTYPE(0)
     RIDPOS(1)  RIDLENG(3)  RIDTYPE(AN)

2W3  DICTIONARYNAME(MMTHL1_DICTIONARY)  RECORDIDINFONAME(MMTHL1_RECORDID)  FORMAT(MMTHL1)
        DESC(MAUI MAP TEST - HL1 - SUW)  ADFTPIDFLD(TRADINGPART) ADFBEGINRECID(MMTHL1REC)

2W5  DICTIONARYNAME(MMTHL1_DICTIONARY)  RECORDIDINFONAME(MMTHL1_RECORDID)
ADFRECORDNAME(MMTHL1REC)
      DESC(MAUI MAP TEST - HL1 - SUW)  ADFRECORDID(001)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(MMTHL1RECID)
DESC(MAUI MAP TEST - HL1 - SUW) DATATYPE(AN)  FLDLEN(3)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(1)
        ADF_RMEM_MEMBERNAME(MMTHL1RECID) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)
2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(TRADINGPART)
DESC(MAUI MAP TEST - HL1 - SUW) DATATYPE(AN)  FLDLEN(18)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(2)
        ADF_RMEM_MEMBERNAME(TRADINGPART) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(ACFIELD) DESC(MAUI MAP TEST - HL1 - SUW)
        DATATYPE(AC)  FLDLEN(6)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(3)
        ADF_RMEM_MEMBERNAME(ACFIELD) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(FILLER) DESC(MAUI MAP TEST - HL1 - SUW)
        DATATYPE(AN)  FLDLEN(1)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(4)
        ADF_RMEM_MEMBERNAME(FILLER) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(ELIGBEGDATE)
DESC(MAUI MAP TEST - HL1 - SUW) DATATYPE(AN)  FLDLEN(8)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(5)
        ADF_RMEM_MEMBERNAME(ELIGBEGDATE) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(ELIGENDDATE) DESC(MAUI MAP TEST - HL1 - SUW)
        DATATYPE(AN)  FLDLEN(8)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(6)
        ADF_RMEM_MEMBERNAME(ELIGENDDATE) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(SUBSCRIBER) DESC(MAUI MAP TEST - HL1 - SUW)
        DATATYPE(AN)  FLDLEN(18)  ADFSENDMAP(0) ADFRECEIVEMAP(0)
```

```
2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(7)
       ADF_RMEM_MEMBERNAME(SUBSCRIBER) ADF_RMEM_MEMBERTYPE(FIELD)  ADF_RMEM_MAXUSE(1)

2W6  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFSTRUCTNAME(FAMILY) DESC(MAUI MAP TEST - HL1 - SUW)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(DEPFIRSTNAME) DESC(MAUI MAP TEST - HL1 - SUW)
       DATATYPE(AN)  FLDLEN(9)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WB  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_SMEM_STRUCTNAME(FAMILY) ADF_SMEM_SEQNUM(1)
       ADF_SMEM_MEMBERNAME(DEPFIRSTNAME) ADF_SMEM_MEMBERTYPE(FIELD)  ADF_SMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(DEPLASTNAME) DESC(MAUI MAP TEST - HL1 - SUW)
       DATATYPE(AN)  FLDLEN(9)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WB  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_SMEM_STRUCTNAME(FAMILY) ADF_SMEM_SEQNUM(2
       ADF_SMEM_MEMBERNAME(DEPLASTNAME) ADF_SMEM_MEMBERTYPE(FIELD)  ADF_SMEM_MAXUSE(1)

2W7  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADFFIELDNAME(DEPDOB) DESC(MAUI MAP TEST - HL1 - SUW)
       DATATYPE(AN)  FLDLEN(8)  ADFSENDMAP(0) ADFRECEIVEMAP(0)

2WB  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_SMEM_STRUCTNAME(FAMILY) ADF_SMEM_SEQNUM(3)
       ADF_SMEM_MEMBERNAME(DEPDOB) ADF_SMEM_MEMBERTYPE(FIELD)  ADF_SMEM_MAXUSE(1)

2WA  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_RMEM_RECORDNAME(MMTHL1REC) ADF_RMEM_SEQNUM(8)
       ADF_RMEM_MEMBERNAME(FAMILY) ADF_RMEM_MEMBERTYPE(STRUCT)  ADF_RMEM_MAXUSE(15)

2W8  DICTIONARYNAME(MMTHL1_DICTIONARY)  ADF_HMEM_ADFNAME(MMTHL1) ADF_HMEM_SEQNUM(1)
       ADF_HMEM_MEMBERNAME(MMTHL1REC) ADF_HMEM_MEMBERTYPE(REC)  ADF_HMEM_AREA(Detail)
       ADF_HMEM_MAXUSE(1)  ADF_HMEM_INFLOOP(0)

000
```

Data format dictionary record (2W1)

This record defines the data format dictionary being used. The fields are described in Table 31.

Table 31. Data format dictionary record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
DESC	34–83	50	Char	Description
CMTTEXT	84–2131	2048	Char	Comments
ALLADFS	2132	1	Char	Include all ADFs

Data format record ID record (2W2)

This record defines the data format record being used. The fields are described in Table 32 on page 247.

Table 32. Data format record ID record

Tag	Position	Length	Type	Description
RECORDIDINFONAME	4–33	30	Char	Record ID information name
DESC	34–83	50	Char	Description
ADFTYPE	84–84	1	Char	Data format type
RIDPOS	85–90	6	Char	Record ID position
RIDLENG	91–101	11	Char	Record ID length
RIDTYPE	102–103	2	Char	Record ID type
CMTTEXT	104–2151	2048	Char	Comments

Data format header record (2W3)

This record defines the data format header being used. The fields are described in Table 33.

Table 33. Data format header record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
RECORDIDINFONAME	34–63	30	Char	Record ID information name
FORMAT	64–79	16	Char	Data format name
DESC	80–129	50	Char	Description
FILEID	130–137	8	Char	logical name
FILETYPE	138–139	2	Char	File type
ADFTPIDFLD	140–169	30	Char	Field containing trading partner ID
ADFBEGINRECID	170–199	30	Char	Field containing beginning record ID
ADFENDRECID	200–229	30	Char	Field containing ending record ID
ADFGENRTCDE	230–259	30	Char	Field containing generic routing code
CMTTEXT	260–2307	2048	Char	Comments
INTSNDQUALFLD	2308–2337	30	Char	Field containing interchange sender ID qualifier
INTSNDIDFLD	2338–2367	30	Char	Field containing interchange sender ID
INTRCVQUALFLD	2368–2397	30	Char	Field containing interchange receiver ID qualifier

Table 33. Data format header record (continued)

Tag	Position	Length	Type	Description
INTRCVFLD	2398–2427	30	Char	Field containing interchange receiver ID
APPLTPIDFLD	2428–2457	30	Char	Field containing application trading partner nickname
EDITPIDFLD	2458–2487	30	Char	Field containing EDI trading partner nickname
ADFCMDELIM	2488	1	Char	Comma-separated value delimiter
ADFRECDELIM	2489–2490	2	Char	Record delimiter
ADFUNICODE	2491–2506	16	Char	Code page
ADFTXTDELIM	2507–2508	2	Char	Reserved for future use
ADFTXTQUAL	2509–2538	30	Char	Reserved for future use

Data format loop record (2W4)

This record defines the data format loop being used. The fields are described in Table 34.

Table 34. Data format loop record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
RECORDIDINFONAME	34–63	30	Char	Record ID information name
ADFLOOPNAME	64–93	30	Char	Loop name
DESC	94–143	50	Char	Description
CMTTEXT	144–2191	2048	Char	Comments

Data format record record (2W5)

This record defines the data format record being used. The fields are described in Table 35.

Table 35. Data format record record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
RECORDIDINFONAME	34–63	30	Char	Record ID information name
ADFRECORDNAME	64–93	30	Char	Record name
DESC	94–143	50	Char	Description

Table 35. Data format record record (continued)

Tag	Position	Length	Type	Description
ADFRECORDID	144–159	16	Char	Record ID
CMTTEXT	160–2207	2048	Char	Comments

Data format structure record (2W6)

This record defines the data format structure being used. The fields are described in Table 36.

Table 36. Data format structure record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
ADSTRUCTNAME	34–63	30	Char	Structure name
DESC	64–113	50	Char	Description
CMTTEXT	114–2161	2048	Char	Comments

Data format field record (2W7)

This record defines the data format fields being used. The fields are described in Table 37.

Table 37. Data format field record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
ADFFIELDNAME	34–63	30	Char	Field name
DESC	64–113	50	Char	Description
DATATYPE	114–115	2	Char	Data type
FLDLEN	116–121	6	Char	Field length
ADFMAPCOMMANDS	122–201	80	Char	Mapping command line
ADFSENDMAP	202	1	Char	Send map
ADFRECEIVEMAP	203	1	Char	Receive map
ADFVALTBLNAME	204–211	8	Char	Code list name
CMTTEXT	212–2259	2048	Char	Comments

Data format header details record (2W8)

This record defines the data format header details being used. The fields are described in Table 38 on page 250.

Table 38. Data format header details record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
ADF_HMEM_ADFNAME	34–49	16	Char	Data format name
ADF_HMEM_SEQNUM	50–59	10	Char	Sequence number
ADF_HMEM_MEMBERNAME	60–89	30	Char	Member name
ADF_HMEM_MEMBERTYPE	90–97	8	Char	Member type
ADF_HMEM_AREA	98–113	16	Char	Area value
ADF_HMEM_MAXUSE	114–118	5	Char	Maximum number of occurrences
ADF_HMEM_INFLOOP	119	1	Char	Infinite loop flag

Data format loop details record (2W9)

This record defines the data format loop details being used. The fields are described in Table 39.

Table 39. Data format loop details record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
ADF_LMEM_LOOPNAME	34–63	30	Char	Loop name
ADF_LMEM_SEQNUM	64–73	10	Char	Sequence number
ADF_LMEM_MEMBERNAME	74–103	30	Char	Member name
ADF_LMEM_MEMBERTYPE	104–111	8	Char	Member type
ADF_LMEM_MAXUSE	112–116	5	Char	Maximum number of occurrences
ADF_LMEM_INFLOOP	117	1	Char	Infinite loop flag

Data format record details record (2WA)

This record defines the data format record details being used. The fields are described in Table 40.

Table 40. Data format record details record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
ADF_RMEM_RECORDNAME	34–63	30	Char	Record name
ADF_RMEM_SEQNUM	64–73	10	Char	Sequence number
ADF_RMEM_MEMBERNAME	74–103	30	Char	Member name
ADF_RMEM_MEMBERTYPE	104–111	8	Char	Member type

Table 40. Data format record details record (continued)

Tag	Position	Length	Type	Description
ADF_RMEM_MAXUSE	112–116	5	Char	Maximum number of occurrences
ADF_RMEM_OCCURSDEPNAME	117–146	30	Char	The field that determines whether this structure should occur

Data format structure details record (2WB)

This record defines the data format structure details being used. The fields are described in Table 41.

Table 41. Data format structure details record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	Dictionary name
ADF_SMEM_STRUCTNAME	34–63	30	Char	Structure name
ADF_SMEM_SEQNUM	64–73	10	Char	Sequence number
ADF_SMEM_MEMBERNAME	74–103	30	Char	Member name
ADF_SMEM_MEMBERTYPE	104–111	8	Char	Member type
ADF_SMEM_MAXUSE	112–116	5	Char	Maximum number of occurrences
ADF_SMEM_OCCUR SDEPNAME	117–146	30	Char	The field that determines whether this structure should occur

Exporting and importing maps

Maps can be exported/imported using tagged or fixed format files. The map set includes:

3T4 Trading partner

3T5 Trading partner send usages/rules

3T6 Trading partner receive usages/rules

3T7 Data transformation map usage/rule

3V1 Map header record

3V2 Map segment record

3V3 Map element record

3V9 Map application control record

3VA Map syntax record

3VB Map local variables record

3VC Map reference record

3VD Map nodes record

3VE Map comments record

BK2 Global variables

There are 20 associated objects for exporting maps. Either maps or usages and rules must be selected.

1. Transaction usages/rules
2. Control string
3. Standard transactions (for DT maps source document definition)
4. Data formats (for DT maps target document definition)umm in
5. Validation tablesTranslation tables
6. User exit routines
7. Trading partner profile
8. Translation exit routines
9. Network security profile
10. Network profile
11. Network commands profile
12. Envelope profile
13. Envelope standard
14. Maps
15. Conversion of prior-release objects
16. Global variables
17. Validation maps
18. FA maps
19. Embedded maps

Note: In trading partner-to-trading partner translations, WebSphere Data Interchange stores control numbers by ISA Sender Qualifier, ISA Receiver Qualifier, and ISA Receiver, ensuring that the receiver gets a contiguous set of control numbers from every sender.

The following example shows an export/import file that contains tags for a send map and the associated send usage/rule.

```
0C1DD5TST1 011025135724ENU010401000000
3V1  TPTID(MMTHL1)  STDID(X12V3R7)  STDTRID(271)  FORMAT(MMTHL1) DIR(S)
GENREQD(N)
      DESC(MAUI MAP TEST - HL1 - SUW Outbound)
DICTIONARYNAME(MMTHL1_DICTIONARY)
      CMPCHK(N)  BASE(T)
3V2  TPTID(MMTHL1)  TABLE(1)  POS(20)  OCCUR(1)  GENFLAG(N)
3V2  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  DEPTHSEQ(1)
PATH(MMTHL1REC)
 HLNODE(1)  HLPARNODE(1)  HL03(20)  GENFLAG(N)
3V2  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(2)  DEPTHSEQ(3)  HLNODE(2)
 HLPARNODE(1)  HL03(21)  PARHL03(20)  GENFLAG(N)
3V2  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(3)  DEPTHSEQ(5)  HLNODE(3)
 HLPARNODE(1)  HL03(22)  PARHL03(21)  GENFLAG(N)
```

```
3V2  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  DEPTHSEQ(6)
PATH(MMTHL1RECFAMILY)  HLNODE(4)  HLPARNODE(1)  HL03(23)  PARHL03(22)
GENFLAG(N)3V2  TPTID(MMTHL1)  TABLE(2)  POS(20)  OCCUR(1)  PARTABLE(2)
 PARPOS(10)
 PAROCCUR(2)  DEPTHSEQ(4)  QUALOCCUR(1)  GENFLAG(N)
3V2  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(1)  PARTABLE(2)  PARPOS(10)
 PAROCCUR(4)  DEPTHSEQ(7)  QUALOCCUR(1)  GENFLAG(N)
3V2  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(2)  PARTABLE(2)  PARPOS(10)
 PAROCCUR(1)  DEPTHSEQ(2)  QUALOCCUR(1)  GENFLAG(N)
3V3  TPTID(MMTHL1)  TABLE(1)  POS(20)  OCCUR(1)  ELEPOS(1)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(0022)
3V3  TPTID(MMTHL1)  TABLE(1)  POS(20)  OCCUR(1)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(11)
3V3  TPTID(MMTHL1)  TABLE(1)  POS(20)  OCCUR(1)  ELEPOS(3)  ELEOCCUR(1)
 PATH(MMTHL1REC)  ADFFIELDNAME(ACFIELD)  DATEEDIT(-4096)
3V3  TPTID(MMTHL1)  TABLE(1)  POS(20)  OCCUR(1)  ELEPOS(4)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&DATE)
3V3  TPTID(MMTHL1)  TABLE(1)  POS(20)  OCCUR(1)  ELEPOS(5)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&TIME)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  ELEPOS(1)  ELEOCCUR(2)
 DATEEDIT(-4096)  LITVAL(&SET MAPIT 0)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  ELEPOS(1)  ELEOCCUR(3)
 PATH(MMTHL1REC)  ADFFIELDNAME(SUBSCRIBER)  DATEEDIT(-4096)
LITVAL(&IFDATA &SET MAPIT 1)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  ELEPOS(1)  ELEOCCUR(4)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HLID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  ELEPOS(2)
 ELEOCCUR(1) DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HLPID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  ELEPOS(3)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HCODE)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(1)  ELEPOS(4)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HCHILD)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(2)  ELEPOS(1)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HLID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(2)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HLPID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(2)  ELEPOS(3)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HCODE)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(2)  ELEPOS(4)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HCHILD)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(3)  ELEPOS(1)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HLID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(3)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HLPID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(3)  ELEPOS(3)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HCODE)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(3)  ELEPOS(4)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&HCHILD)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  ELEPOS(1)  ELEOCCUR(2)
 DATEEDIT(-4096)  LITVAL(&SET MAPIT 0)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  ELEPOS(1)  ELEOCCUR(3)
 PATH(MMTHL1RECFAMILY)  ADFFIELDNAME(DEPDOB)  DATEEDIT(-4096)
 LITVAL(&IFDATA &SET MAPIT 1)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  ELEPOS(1)  ELEOCCUR(4)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HLID)
```

```
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HLPID)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  ELEPOS(3)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HCODE)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(10)  OCCUR(4)  ELEPOS(4)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&IF(MAPIT = 1?) &HCHILD)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(20)  OCCUR(1)  ELEPOS(1)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(AC)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(20)  OCCUR(1)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(&ACFIELD)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(1)  ELEPOS(1)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(DD)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(1)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(D)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(1)  ELEPOS(3)  ELEOCCUR(1)
 PATH(MMTHL1RECFAMILY)  ADFFIELDNAME(DEPLASTNAME)  DATEEDIT(-4096)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(1)  ELEPOS(4)  ELEOCCUR(1)
 PATH(MMTHL1RECFAMILY)  ADFFIELDNAME(DEPFIRSTNAME)  DATEEDIT(-4096)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(2)  ELEPOS(1)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(SS)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(2)  ELEPOS(2)  ELEOCCUR(1)
 DATEEDIT(-4096)  LITVAL(S)
3V3  TPTID(MMTHL1)  TABLE(2)  POS(30)  OCCUR(2)  ELEPOS(3)  ELEOCCUR(1)
 PATH(MMTHL1REC)  ADFFIELDNAME(SUBSCRIBER)  DATEEDIT(-4096)
000
```

The following tables define the fields for exporting/importing maps.

Map header record (3V1)

This record defines the map header being used. The fields are described in Table 42.

Table 42. Map header record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
STDID	20–49	30	Char	EDI standard dictionary name
STDTRID	50–57	8	Char	EDI standard transaction ID
FORMAT	58–73	16	Char	Data format ID
DIR	74–74	1	Char	Map type (send/receive/data transformation) flag
GENREQD	75–75	1	Char	Compile required
DESC	76–125	50	Char	Description
DICTIONARYNAME	126–155	30	Char	Data format dictionary name
CMPCHK	156	1	Char	Compliance check flag
BASE	157	1	Char	Future use

Map segment record (3V2)

This record defines the map segment being used. The fields are described in Table 43.

Table 43. Map segment record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
TABLE	20	1	Char	Table
POS	21–25	5	Char	Position in table
OCCUR	26–30	5	Char	Occurrence
PARTABLE	31	1	Char	Parent table
PARPOS	32–36	5	Char	Parent position
PAROCCUR	37–41	5	Char	Parent occurrence
DEPTHSEQ	42–46	5	Char	Depth sequence number
QUALOCCUR	47–52	6	Char	Qualification occurrence number
QUALELEPOS	53–56	4	Char	Qualified element position
QUALELEREPEATNO	57–61	5	Char	Qualified element repeat number
QUALSUBELEPOS	62–65	4	Char	Qualified sub-element position
PATH	66–320	255	Char	Qualified path
QUALTEXT	321–365	45	Char	Element loop value
HLNODE	366–370	5	Char	Hierarchical loop node number
HLPARNODE	371–375	5	Char	Hierarchical loop parent node number
HL03	376–377	2	Char	HL03 value
PARHL03	378–379	2	Char	HL03 parent value
FLDHL03	380–409	30	Char	HL03 field value
PARFLDHL03	410–439	30	Char	HL03 field parent value
GENFLAG	440	1	Char	Generate segment
DOSNIPET	441	1	Char	Snipet text
BSNIPET	442–521	80	Char	Before snipet flag
ASNIPET	522–601	80	Char	After snipet flag
CMTTEXT	602–2649	2048	Char	Segment mapping comments

Map element record (3V3)

This record defines the map element being used. The fields are described in Table 44.

Table 44. Map element record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name

Table 44. Map element record (continued)

Tag	Position	Length	Type	Description
TABLE	20	1	Char	Table ID
POS	21–25	5	Char	Position in table
OCCUR	26–30	5	Char	Table occurrence number
ELEPOS	31–35	5	Char	Element position number
SUBELEPOS	36–40	5	Char	Sub-element position number
ELEOCCUR	41–45	5	Char	Element occurrence number
PATH	46–300	255	Char	Path name
ADFFIELDNAME	301–330	30	Char	Data format field name
ELEQUALPOS	331–335	5	Char	Qualifying element's position number
SUBELEQUALPOS	336–340	5	Char	Qualifying sub-element's position number
ELEQUALOCCUR	341–345	5	Char	Qualifying element's occurrence number
QUALTEXT	346–386	45	Char	Element qualifier
TRNTAB	391–398	8	Char	Translation table
USEREXIT	699–406	8	Char	User exit ID
VALTAB	407–414	8	Char	Code list ID
DATEEDIT	415–416	2	Char	Date edit field
SUBPOS	417–422	6	Char	Substring or concatenation position
SUBLEN	423–428	6	Char	Substring or concatenation length
ACCUM1	429–430	2	Char	Accumlator1
ACTION1	431–432	2	Char	Accumlator1 action
ACCUM2	433–434	2	Char	Accumlator2
ACTION2	435–436	2	Char	Accumlator2 action
ACCUM3	437–438	2	Char	Accumlator3
ACTION3	439–440	2	Char	Accumlator3 action
ACCUM4	441–442	2	Char	Accumlator4
ACTION4	443–444	2	Char	Accumlator4 action
LITVAL	445–524	80	Char	Code Snipet
CMTTEXT	525–2572	2048	Char	Element comments
ELEREPEATNO	2573–2577	5	Char	Element repeat occurrence number
QUALPATH	2578–2832	255	Char	Qualifying path
UNIQUEID	2833–2838	6	Char	Mapping unique ID

Map application control record (3V9)

This record defines the map application control being used. The fields are described in Table 45.

Table 45. Map application control record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
UNIQUEID	20	1	Char	Unique ID
SEQUENCE	21	1	Char	Sequence
PATH	22–276	255	Char	Path name
ADFFIELDNAME	277–306	30	Char	Data format field name
LENGTH	307–311	5	Char	Field length

Map syntax record (3VA)

This record defines the map syntax being used. The fields are described in Table 46.

Table 46. Map syntax record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
UNIQUEID	20–24	5	Char	Unique ID
BASE	25	1	Char	Future use
SEQNO	26–30	6	Char	Sequence number
SYNTAX	31	1	Char	Syntax type
DICTIONARYNAME	32–61	30	Char	Dictionary name
DOCNAME	62–91	30	Char	Document definition name

Map local variables record (3VB)

This record defines the map local variables being used. The fields are described in Table 47.

Table 47. Map local variables record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
VARNAME	20–49	30	Char	Variable name
DISPNAME	50–79	30	Char	Displayed variable name
DESC	80–129	50	Char	Description
DATATYPE	130	1	Char	Data type
SCOPE	131	1	Char	Scope
DEMAX	132–136	5	Char	Maximum length
INITVAL	137–168	32	Char	Initial value

Map reference record (3VC)

This record defines the map reference being used. The fields are described in Table 48.

Table 48. Map reference record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
UNIQUEID	20–25	6	Char	Unique ID
TYPE	26	1	Char	Type
PARENTID	27–37	11	Char	Syntax ID
SYNTAX	38	1	Char	Syntax type
DICTIONARYNAME	39–68	30	Char	Dictionary name
FORMAT	69–98	30	Char	Document definition name
PATH	99–353	255	Char	Path

Map nodes record (3VD)

This record defines the map nodes being used. The fields are described in Table 49.

Table 49. Map nodes record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
UNIQUEID	20–25	6	Char	Unique ID
PARENTID	26–36	11	Char	Parent's unique ID
SEQNO	37–47	11	Char	Sequence

Table 49. Map nodes record (continued)

Tag	Position	Length	Type	Description
TYPE	48	1	Char	Map type
SUBTYPE	49	1	Char	Map sub type
ASSOCFWD	50–55	6	Char	Associated forward
ASSOCBKD	56–61	6	Char	Associated backward
POSNO	62–70	9	Char	Position number
EDIT	71–71	1	Char	Map edit
REFID	72–77	6	Char	Map reference ID
REPEATS	78–78	1	Char	Node repeats
QUALIFIED	79–79	1	Char	Qualified flag
CMTTEXT	80–2128	2048	Char	Comment

Map commands record (3VE)

This record defines the map commands being used. The fields are described in Table 50.

Table 50. Map commands record

Tag	Position	Length	Type	Description
TPTID	4–19	16	Char	Map name
UNIQUEID	20–25	6	Char	Unique ID
PARENTID	26–36	11	Char	Node's unique ID
SEQNO	37–47	11	Char	Sequence
VERSION	48–53	6	Char	Version
COMMAND	54–2101	2048	Char	Map command

Global variables details record (BK2)

This record defines the global variables being used. The fields are described in Table 51.

Table 51. Global variables details record

Tag	Position	Length	Type	Description
VARNAME	4–33	30	Char	Variable name
DISPNAME	34–63	30	Char	Displayed variable name
DESC	64–113	50	Char	Description
DATATYPE	114	1	Char	Data type
SCOPE	115	1	Char	Scope
DEMAX	116–120	5	Char	Maximum length

Table 51. Global variables details record (continued)

INITVAL	121–152	32	Char	Initial value

Send usage record (3T5)

This record defines the send usage/rule being used. The fields are described in Table 52.

Table 52. Trading partner send usage record

Tag	Position	Length	Type	Description
INTPID	4–38	35	Char	Internal trading partner ID
FORMAT	39–54	16	Char	Data format ID
TPTID	55–70	16	Char	Trading partner mapping transaction ID
TPNICKN	71–86	16	Char	Trading partner nickname
ENVTYPE	87–87	1	Char	EDI standard envelope type
STDPROF	88–95	8	Char	EDI standard profile name
APPSNDID	96–130	35	Char	Application sender ID
APPRECID	131–165	35	Char	Application receiver ID
APPASSWD	166–179	14	Char	Application password
PSTTRXIT	180–187	8	Char	Post-translation exit routine name
ACTIVE	188	1	Char	Active usage for this INTPID/FORMAT combination
ACKREQ	189	1	Char	Acknowledgment expected flag
TESTIND	190	1	Char	Test indicator
LOGAPP	191	1	Char	Log application data
ENFORCEHIER	192	1	Char	Enforce structure hierarchy
STRUCTDATACHK	193	1	Char	Structure must produce data
TRVALVL	194	1	Char	Validation level
TRERLVL	195	1	Char	Acceptable error level
GRPSECPR	196–203	8	Char	Group network security profile name
GRPENCKY	204–219	16	Char	Group encryption key name
GRPAUTKY	220–235	16	Char	Group authentication key name
TRNSECPR	236–243	8	Char	Trans network security profile name
TRNENCKY	244–259	16	Char	Trans encryption key name
TRNAUTKY	260–275	16	Char	Trans authentication key name
APPLTPID	276–291	16	Char	Application trading partner nickname

Table 52. Trading partner send usage record (continued)

Tag	Position	Length	Type	Description
ALPHANUM	292–299	8	Char	Alternate table for ALPHANUM
CHARSET	300–307	8	Char	Alternate table for CHARSET
CNTRX	308	1	Char	Control numbers utilization flag

Receive usage record (3T6)

This record defines the receive usage/rule being used. The fields are described in Table 53.

Table 53. Trading partner receive usage record

Tag	Position	Length	Type	Description
TPNICKN	4–19	16	Char	Trading partner nickname
STDTRID	20–27	8	Char	EDI standard transaction ID
TPTID	28–43	16	Char	Map name
APSNDRCV	44–78	35	Char	Application sender/receiver ID
AGENCY	79–86	8	Char	Responsible agency code
VER	87–94	8	Char	Version
REL	95–102	8	Char	Release
SNDRCVFL	103	1	Char	Sender or receiver flag
INTPID	104–138	35	Char	Internal trading partner ID
STDPROF	139–146	8	Char	Standard profile name
APPASSWD	147–160	14	Char	Application password
PRETRXIT	161–168	8	Char	Pre-translation exit routine
FILEID	169–176	8	Char	Logical receiving data set name
FILETYPE	177–178	2	Char	Type of receive file
ACTIVE	179	1	Char	Active usage flag for this trading partner
ACKTRN	180–187	8	Char	Acknowledgment transaction ID
TESTIND	188	1	Char	Test indicator
LOGAPP	189	1	Char	Log application data
OVERLAYCHK	190	1	Char	Consider data overlay an error
UNEXPFLDCHK	191	1	Char	Consider unexpected field an error
UNEXPSEGCHK	192	1	Char	Consider unexpected segment an error
SWITCHROUTING	193	1	Char	Switch application routing IDs on functional acknowledgment
TRVALVL	194	1	Char	Validation level

Table 53. Trading partner receive usage record (continued)

Tag	Position	Length	Type	Description
TRERLVL	195	1	Char	Acceptable error level
INTFA	196	1	Char	Internal functional acknowledgment switch
APPLTPID	197–212	16	Char	Application trading partner nickname
ALPHANUM	213–220	8	Char	Alternate table for ALPHANUM
CHARSET	221–228	8	Char	Alternate table for CHARSET
CNTRX	229	1	Char	Control numbers utilization flag

Data transformation map rule record (3T7)

This record defines the data transformation map being used. The fields are described in Table 54.

Table 54. Data transformation map rule record

Tag	Position	Length	Type	Description
MAPID	4–20	16	Char	Map name
DOCID	21–49	30	Char	Document definition name
DICTIONARYNAME	50–79	30	Char	Dictionary name
DESCRIPT	80–109	30	Char	Description
PROCESS	110–149	40	Char	Process ID
SENDTPID	150–165	16	Char	Sending trading partner nickname
RECVTPID	166–181	16	Char	Receiving trading partner nicknamme
TESTIND	182	1	Char	Test indicator
ENVTYPE	183	1	Char	EDI standard envelope type
STDPROF	184–191	8	Char	EDI standard profile name
APPSNDID	192–226	35	Char	Application sender ID
APPRECID	227–261	35	Char	Application receiver ID
APPASSWD	262–275	14	Char	Application password
SYNTAX	276–278	3	Char	Syntax type
ACTIVE	279	1	Char	Active usage for this INTPID/FORMAT combination
ACKREQ	280	1	Char	Acknowledgment expected flag
INBVALLVL	281	1	Char	Inbound validation level
OUTBVALLVL	282	1	Char	Outbound validation level
FAENVTYPE	283	1	Char	Functional acknowledgment envelope type

Table 54. Data transformation map rule record (continued)

Tag	Position	Length	Type	Description
FAENVPROF	284–291	8	Char	Functional acknowledgment envelope profile name
GRPLVLFA	292	1	Char	Group level functional acknowledgment
TRERLVL	293	1	Char	Acceptable error level
INBALPHANUM	294–301	8	Char	Inbound alternate table for ALPHANUM
INBCHARSET	302–309	8	Char	Inbound alternate table for CHARSET
OUTBALPHANUM	310–317	8	Char	Outbound alternate table for ALPHANUM
OUTBCHARSET	318–325	8	Char	Outbound alternate table for CHARSET
CNTRX	326	1	Char	Control numbers utilization flag
FILEID	327–334	8	Char	Logical application data set name
FILETYPE	335–336	2	Char	Type of application file
ACKDOCID	337–366	30	Char	Acknowledgment document ID
INBVALMAP	367–396	30	Char	Inbound validation map ID
OUTBVALMAP	397–426	30	Char	Outbound validation map ID

Exporting and importing table definitions

Table definitions can be exported/imported using either tagged or fixed format files. The table definitions set includes:

B1 Table definitions

B2 Table entry

The following example shows an export/import file that contains tags for table definitions.

```
0C1SMITH   940419161616ENU010103000000

8B1TBLID(AAAAA)  DATECRT(940325)  TBLDSC(dd)  PUBAUTH(ALL)  FLAGS(00)

8B2TBLID(AAAAA)  VAR1(A)

000
```

The following tables define the fields for exporting/importing tables.

Export/Import table definition (B1)

Table 55 defines the fields for exporting and importing table definitions.

Table 55. Table definition fields for export and import

Tag	Position	Length	Type	Description
TBLID	4–11	8	Char	Table ID
DATECRT	12–17	6	Char	Date created
TBLDSC	18–52	35	Char	Table description
PUBAUTH	53–58	6	Char	Authority
OWNER	59–66	8	Char	Owner
FLAGS	67–68	2	Char	Flags (always **00**)
KEYID	69–86	18	Char	Key field ID
KEYLN	87–92	6	Char	Key field length
RECLN	93–98	6	Char	Record length
NUMFLDS	99–104	6	Char	Number of fields in a table row
TBLTYPE	105	1	Char	Table type
FLD1ID	106–123	18	Char	Field 1 ID
FLD1OFF	124–129	6	Char	Field 1 offset
FLD1LEN	130–132	3	Char	Field 1 length
FLD1TYPE	133–134	2	Char	Field 1 data type
FLD1DSC	135–169	35	Char	Field 1 description
FLD2ID	170–187	18	Char	Field 2 ID
FLD2OFF	188–193	6	Char	Field 2 offset
FLD2LEN	194–196	3	Char	Field 2 length
FLD2TYPE	197–198	2	Char	Field 2 data type
FLD2DSC	199–233	35	Char	Field 2 description
TAG	234–241	8	Char	Future index tag

Export/Import table definition (B1) field descriptions

Table ID (TBLID): A name for referring to the table.

Date created (DATECRT): The date this table was created.

Table description (TBLDSC): A brief description of this table.

Authority (PUBAUTH): For WebSphere Data Interchange use only.

Owner (OWNER): The user ID of the owner of this table.

Flags (FLAGS): Always set to 00.

Key ID (KEYID): The ID of the field used as the key for performing lookups. Validation and type **T** translation tables use the first (or only) field as the key field. Type **R** translation tables use the second field.

Key field length (KEYLN): The length of the key field.

Record length (RECLN): The length of a row in this table.

Number of fields (NUMFLDS): The total number of fields in a row in this table. Validation tables have one field. Translation tables have two.

Table type (TBLTYPE): The type of table. Valid values are:

T Translation table. Contains pairs of values: the value to be translated and the value to be substituted for it. For example, you can translate a purchaser's part number to a supplier's part number before sending a purchase order. The table is organized with the application value as the key value and the EDI standard or trading partner value as data. With this type of table, multiple application values can translate to the same EDI standard value, but each EDI standard value translates to one and only one application value. This table gives the best performance when translating from an application to an EDI standard.

R Translation table. Serves the same purpose as the type **T** table but is organized with the EDI standard value as the key value and the application value as data. With this type of table, multiple EDI standard values can translate to the same application value, but each application value translates to one and only one EDI standard value. This table gives the best performance when translating from an EDI standard to an application.

V Validation table. This table contains a list of acceptable values for a field. It can contain several values or only one value.

Field 1 ID (FLD1ID): The ID of the first column in a row of this table.

Field 1 offset (FLD1OFF): The offset into a row of this table where the first field begins.

Field 1 length (FLD1LEN): The maximum number of characters one entry in this column contains. The maximum is **35**. For translation tables (data type T) the maximum length for Field 1 and FIeld 2 combined is 68 characters. Include decimal points and integer signs as part of the data length. For example, the value -12.34 requires a length of 6.

Field 1 data type (FLD1TYPE): The data type for entries in this column. Valid values are:

CH Character. Any combination of characters.

R Real. A numeric field that requires a decimal point for fractional values. The decimal point is optional for integers. A sign (+ or -) is optional.

Field 1 description (FLD1DSC): A brief description of this field.

Field 2 ID (FLD2ID): The ID of the second column in a row of this table (if there is one).

Field 2 offset (FLD2OFF): The offset into a row of this table where the second field begins.

Field 2 length (FLD2LEN): The maximum number of characters one entry in this column contains. The maximum is **35**. For translation tables (data type T) the maximum length for Field 1 and FIeld 2 combined is 68 characters. Include decimal points and integer signs as part of the data length. For example, the value -12.34 requires a length of 6.

Field 2 data type (FLD2TYPE): The data type for entries in this column. Valid values are:

CH Character. Any combination of characters.

R Real. A numeric field that requires a decimal point for fractional values. The decimal point is optional for integers. A sign (+ or -) is optional.

Field 2 description (FLD2DSC): A brief description of this field.

Tag (TAG): Reserved for future use.

Export/Import table entry (B2)

Table 56 defines the tags and lengths for exporting/importing table entries.

Table 56. Table entry export import tags and lengths

Tag	Position	Length	Type	Description
TBLID	4–11	8	Char	Table ID
VAR1	12-Variable	Variable	Char	Translate FROM field value
VAR2	Variable	Variable	Char	Translate TO field value

Table entry (B2) field descriptions

Table ID (TBLID): A name for referring to the table.

VAR1: For validation tables, the valid value to be entered into the table. For translation tables, the application value (user-defined) to be associated with a value from the EDI standard.

VAR2: Applies only for translation tables. The EDI standard value to be associated with an application value (user-defined).

Exporting and importing XML records

XML data definitions can be exported/imported using either tagged or fixed format files. The XML definitions set includes:

AJ1 XML dictionary record

AJ2 XML DTD header record

AJ3 XML DTD detail record

The following example shows an export/import file that contains tags for XML definitions.

```
0C1DD5TST1 011025135847ENU010401000000
AJ1  DICTIONARYNAME(TESTS)  DESC(Test DTDs)  ALLDTDS(Y)
AJ2  DICTIONARYNAME(TESTS)  DOCNAME(POXML5SR)
     DESC(Order with Sender/Receiver)
     DTDROOTELEM(OrderSR)  DTDSENDQUALELEM(OrderSRHeaderSenderQualifier)
     DTDSENDIDELEM(OrderSRHeaderSenderId)
     DTDRECVQUALELEM(OrderSRHeaderReceiverQualifier)
     DTDRECVIDELEM(OrderSRHeaderReceiverId)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(1) DTDDATA(<??xml
     version="1.0" encoding="ISO-8859-1"??>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(2)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(3) DTDDATA(<!-- @version: -->)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(4)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(5) DTDDATA(<!ELEMENT
     OrderSR (Header, DetailLoop*, Trailer?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(6)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(7) DTDDATA(<!ELEMENT Header
      (PONum, PODate, Sender, Receiver?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(8)
DTDDATA(<!ATTLIST Header typecode CDATA #REQUIRED>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(9)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(10)
DTDDATA(<!ELEMENT PONum (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(11) DTDDATA(<!ELEMENT PODate (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(12)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(13)
DTDDATA(<!ELEMENT Sender (Id, Qualifier?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(14)
DTDDATA(<!ELEMENT Receiver (Id, Qualifier?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(15)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(16) DTDDATA(<!ELEMENT Id (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(17) DTDDATA(<!ELEMENT Qualifier (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(18)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(19) DTDDATA(<!ELEMENT
     DetailLoop (ItemNumber, SubDetail+?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(20)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(21)
DTDDATA(<!ELEMENT ItemNumber (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(22)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(23) DTDDATA(<!ELEMENT SubDetail
     (Description, Quantity, UnitPrice?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(24)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(25)
DTDDATA(<!ELEMENT Description (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(26) DTDDATA(<!ELEMENT Quantity (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(27) DTDDATA(<!ELEMENT UnitPrice (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(28)
```

```
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(29) DTDDATA(<!ELEMENT
     Trailer (ItemCount, TotalBucks?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(30)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(31) DTDDATA(<!ELEMENT ItemCount (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(32)
DTDDATA(<!ELEMENT TotalBucks (#PCDATA?)>)
AJ3  DICTIONARYNAME(TESTS) DOCNAME(POXML5SR) SEQNO(33)
000
```

The following tables define the fields for exporting/importing XML definitions.

XML dictionary record (AJ1)

This record defines the XML dictionary being used. The fields are described in Table 57.

Table 57. XML dictionary record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	XML dictionary name
DESC	34–83	50	Char	Description
ALLDTDS	84	1	Char	All XML DTDs flag

XML DTD header record (AJ2)

This record defines the XML DTD header being used. The fields are described in Table 58.

Table 58. XML DTD header record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	XML dictionary name
DOCNAME	34–63	30	Char	XML DTD Name
DESC	64–113	50	Char	Description
DTDROOTELEM	114–177	64	Char	Root element
DTDSENDQUALELEM	178–432	255	Char	Element containing sender qualifier
DTDSENDIDELEM	433–687	255	Char	Element containing sender ID
DTDSENDALIASTBL	688–695	8	Char	Sender alias table
DTDRECVQUALELEM	696–950	255	Char	Element containing receiver qualifier
DTDRECVIDELEM	951–1205	255	Char	Element containing receiver ID
DTDRECVALIASTBL	1206–1213	8	Char	Receiver alias table
DTDLOC	1214–1363	150	Char	Source location of DTD

XML DTD details record (AJ3)

This record defines the XML DTD details being used. The fields are described in Table 59 on page 269.

Table 59. XML DTD details record

Tag	Position	Length	Type	Description
DICTIONARYNAME	4–33	30	Char	XML dictionary name
DOCNAME	34–63	30	Char	XML DTD Name
SEQNO	64–74	11	Char	Data sequence number
DTDCONT	75	1	Char	Continuation flag
DTDDATA	76–330	255	Char	DTD data

Additional profile layouts

The sixteen additional profile layouts defined in the following tables can be exported or imported using either tagged or fixed format files. These profile layouts are:

- "Mailbox (requestor) profile (REQPROF-P2)"
- "Network security profile (SECUPROF-P2)" on page 270
- "Network profile (NETPROF-P2)" on page 271
- "Network commands profile (NETOP-P2)" on page 272
- "Application defaults profile (APPDEFS-P2)" on page 272
- "User exit profile (ADAMCTL-P2)" on page 273
- "Language profile (LANGPROF-P2)" on page 273
- "EDIFACT (E envelope) profile (E-P2)" on page 275
- "ICS (I envelope) profile (I-P2)" on page 276
- "UN/TDI (T envelope) profile (T-P2)" on page 277
- "UCS (U envelope) profile (U-P2)" on page 279
- "X12 (X envelope) profile (X-P2)" on page 279
- "Continuous receive profile (CONTRECV-P2 for CICS only)" on page 280
- "CICS performance profile (SYSPROF-P2 for CICS only)" on page 281
- "WebSphere MQ queue profile (WebSphere MQ-P2)" on page 281

Mailbox (requestor) profile (REQPROF-P2)

This record defines the mailbox (requestor) settings being used. The fields are described in Table 60.

Table 60. Mailbox (requestor) profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **REQPROF**
REQID	12–27	16	Char	Mailbox (requestor) ID
NETID	28–35	8	Char	Network ID
ACCTID	36–67	32	Char	Network account number
USERID	68–99	32	Char	Network user ID
NETPASS	100–115	16	Char	Network password
MSGCLS	116–123	8	Char	Network message user class
FILEID	124–131	8	Char	Transaction receive logical name
NETCLS	132	1	Char	Network classification

Table 60. Mailbox (requestor) profile (continued)

Tag	Position	Length	Type	Description
NETCHG	133	1	Char	Network charges code
NETACK	134	1	Char	Network acknowledgment code
NETVCHK	135	1	Char	Validate destination
NETRETN	136–138	3	Char	Message retention
NETEDIO	139	1	Char	EDI processing option
NETEDIP	140	1	Char	EDI processing override
STGFMTOV	141	1	Char	Storage format override
STGFMT	142	1	Char	Storage format
NETCMDS	143–150	8	Char	Member name of network commands file
TIMEOUT	151–154	4	Dec	Command line timeout value
NETACKPG	155–162	8	Char	Program to handle network acknowledgments from remote network
ALTNETPH	163–194	32	Char	Alternate dial connection phone number
COMPRESS	195	1	Char	Compression
PRIORITY	196	1	Char	Delivery priority
DESCRIPT	197–226	30	Char	Member description

Network security profile (SECUPROF-P2)

This record defines the network security settings being used. The fields are described in Table 61.

Table 61. Network security profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **SECUPROF**
SECID	12–19	8	Char	Security ID
SECORIGN	20–35	16	Char	Security originator
SECRECIP	36–51	16	Char	Security recipient
AUTTYPE	52	1	Char	Authorization type
AUTCODE	53	1	Char	Authorization code
ENCTYPE	54	1	Char	Encryption code
FILTYPE	55	1	Char	Filtering type
ENCPROG	56–63	8	Char	Encryption program
AUTPROG	64–71	8	Char	Authentication program
COMPROG	72–79	8	Char	Compression program

Table 61. Network security profile (continued)

Tag	Position	Length	Type	Description
FILPROG	80–87	8	Char	Filtering program
BUFSIZE	88–92	5	Dec	Buffer size
DESCRIPT	93–122	30	Char	Description

Network profile (NETPROF-P2)

This record defines the network settings being used. The fields are described in Table 62.

Table 62. Network profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **NETPROF**
NETID	12–19	8	Char	Network ID
NETNAME	20–49	30	Char	Network name
COMMRTN	50–57	8	Char	Communication routine name
NETPROG	58–65	8	Char	Network send/receive program name
NETPARM	66–122	57	Char	Network program parameters
CMDINDD	123–130	8	Char	Network program command input file name
CMDRECLN	131–134	4	Dec	Network command record length
QFILEDD	135–142	8	Char	TD queue file name
QRECLEN	143–146	4	Dec	Transaction data record length
TIMEZONE	147–151	5	Char	Time zone
SYSTYPE	152–159	8	Char	System type
SYSLVL	160–163	4	Char	System level
MSGTXTH	164	1	Char	Message text header character
CMDOUTDD	165–172	8	Char	Network program command output file name
MSGHNDLR	173–180	8	Char	Program to process messages
NETSEQ	181–185	5	Char	Sequence number for network
NETAKFILE	186–193	8	Char	File for network acknowledgements
NETPHONE	194–225	32	Char	Dial connection phone number
SCRIPT	226–233	8	Char	Communication script name
DESCRIPT	234–263	30	Char	Description

Network commands profile (NETOP-P2)

This record defines the network commands being used. The fields are described in Table 63.

Table 63. Network operations profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **NETOP**
NETID	12–19	8	Char	Network ID
NETOP	20–27	8	Char	Network command
NETFS	28–35	8	Char	Operation sequence number
BLKNM	36–43	8	Char	Block name
BLKPOS	44–47	4	Dec	Block position
CMDSEQ	48–51	4	Dec	Command line sequence number
CMDPOS	52–55	4	Dec	Command field position
CMDLEN	56–59	4	Dec	Command field length
CMDVAL	60–117	58	Char	Command field value
DESCRIPT	118–147	30	Char	Description

Application defaults profile (APPDEFS-P2)

This record defines the application default settings being used. The fields are described in Table 64.

Table 64. Application defaults profile

Tag	Position	Length	Type	Description
PROFID	0-7	8	Char	Profile ID = **APPDEFS**
APPLICID	8-15	8	Char	Application ID
MEMBER	16-23	8	Char	ACTLOGS profile name
MRFLAG	24	1	Char	Management reporting active
TSFLAG	25	1	Char	Document Store active
TIFLAG	26	1	Char	Transaction image wanted
MONITOR	27-34	8	Char	CICS performance monitor exit name
USFLAG	35	1	Char	Test transaction with production usage
LOGENV	36	1	Char	Log EDI standard data
FAIFLAG	37	1	Char	Functional acknowledgment image
ELFLAG	38	1	Char	Envelope log active
DESCRIPT	39-68	30	Char	Description
ALPHANUM	69-76	8	Char	Name of override ALPHANUM table
CHARSET	77-84	8	Char	Name of override CHARSET table
CTRLYY	85-86	2	Char	Century control year

Table 64. Application defaults profile (continued)

Tag	Position	Length	Type	Description
ELOGSEV	87-88	2	Char	Event log severity threshold
PFILSEV	89-90	2	Char	Print file severity threshold
XFILSEV	91-92	2	Char	XML print file severity threshold
DFILSEV	93-94	2	Char	DF print file severity threshold
LOGASYNC	95	1	Char	Write event log asynchronously
XPRTFLG	96	1	Char	XML print file flag
APRTFLG	97	1	Char	DF print file flag
ENDMFLAG	98	1	Char	End transformation status message
ACCGROUP	99-128	30	Char	Access group
FOLDER	129	64	Char	Folder

User exit profile (ADAMCTL-P2)

This record defines the user exits being used. The fields are described in Table 65.

Table 65. User exit profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **ADAMCTL**
LOGPGNM	12–19	8	Char	Logical name of program
LOADMOD	20–27	8	Char	Load module name
LANG	28	1	Char	Language used in user program
ADUEFLD	29	1	Char	Field exit type
ADUEPST	30	1	Char	Post-translate exit type
ADUEPRE	31	1	Char	Pre-translate exit type
ADUEENC	32	1	Char	Encryption exit type
ADUEAUT	33	1	Char	Authentication exit type
ADUECMP	34	1	Char	Compression exit type
ADUEFLT	35	1	Char	Filtering exit type
ADUEMNT	36	1	Char	Monitor exit type
ADUECOM	37	1	Char	Communication exit type
ADUEMSG	38	1	Char	Message process exit type
ADUEPTP	39	1	Char	Point-to-point exit type
ADUEENV	40	1	Char	Envelope exit type
DESCRIPT	41–70	30	Char	Description

Language profile (LANGPROF-P2)

This record defines the language settings being used. The fields are described in Table 66 on page 274.

Table 66. Language profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **LANGPROF**
LANGID	12–17	6	Char	Language ID
CPAGE	18–22	5	Char	Code page ID
DMASK	23–32	10	Char	Date edit/display mask
TMASK	33–40	8	Char	Time edit/display mask
DECNOT	41	1	Char	Edit/display decimal notation
SIGN	42–43	2	Char	Preferred negative sign (display)
FOLD	44	1	Char	Substitute for non-displayed character
DESCRIPT	45–74	30	Char	Description

EDIFACT (E envelope) profile (E-P2)

This record defines the EDIFACT envelope settings being used. The fields are described in Table 67.

Table 67. EDIFACT (E envelope) profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **E**
EDIFKEY	12–19	8	Char	Profile name
UNB01	20–23	4	Char	Syntax identifier
UNB02	24	1	Char	Syntax version number
UNB02A	25–30	6	Char	Service code list directory version number
UNB02B	31–33	3	Char	Character encoding, coded
UNB02C	34–35	2	Char	Syntax release number
UNB03	36–70	35	Char	Interchange sender ID
UNB04	71–74	4	Char	Sender id qualifier
UNB05	75–109	35	Char	Interchange sender internal ID
UNB05A	110–144	35	Char	Interchange sender internal sub-identification
UNB06	145–179	35	Char	Interchange receiver ID
UNB07	180–183	4	Char	Receiver id qualifier
UNB08	184–218	35	Char	Interchange receiver internal ID
UNB08A	219–253	35	Char	Interchange receiver internal sub-identification
UNB12	280–293	14	Char	Recipient reference/password
UNB13	294–295	2	Char	Recipient reference/password qualifier
UNB14	296–309	14	Char	Application reference
UNB15	310	1	Char	Processing priority code
UNB16	311	1	Char	Acknowledgement request
UNB17	312–346	35	Char	Interchange agreement identifier
UNG01	348–352	6	Char	Message group ID
UNG02	354–387	35	Char	Application sender ID
UNG03	389–391	4	Char	Application sender ID qualifier
UNG04	393–426	35	Char	Application receiver ID
UNG05	427–430	4	Char	Application receiver ID qualifier
UNG09	431–433	3	Char	Controlling agency
UNG10	434–436	3	Char	Message version number
UNG11	437–439	3	Char	Message release number
UNG12	440–445	6	Char	Association assigned code
UNG13	446–459	14	Char	Application password

Table 67. EDIFACT (E envelope) profile (continued)

Tag	Position	Length	Type	Description
UNH04	460–462	3	Char	Message release number
UNH05	463–465	3	Char	Controlling agency
UNH06	466–471	6	Char	Association assigned code
UNH06A	472–477	6	Char	Code list directory version number
UNH06B	476–483	6	Char	Message type sub-function ID
UNH07	484–518	35	Char	Common access reference
UNH08	519–520	2	Char	Sequence of transfers
UNH09	521	1	Char	First and last transfer
UNH10A	522–535	14	Char	Message subset ID
UNH10B	536–538	3	Char	Message subset version number
UNH10C	539–541	3	Char	Message subset release number
UNH10D	542–544	3	Char	Controlling agency
UNH11A	545–558	14	Char	Message implementation guideline ID
UNH11B	559–561	3	Char	Message implementation guideline version number
UNH11C	562–564	3	Char	Message implementation guideline release number
UNH11D	565–567	3	Char	Controlling agency
UNH12A	568–581	14	Char	Scenario ID
UNH12B	582–584	3	Char	Scenario version number
UNH12C	585–587	3	Char	Scenario release number
UNH12D	588–590	3	Char	Controlling agency
DESCRIPT	591–620	30	Char	Description

ICS (I envelope) profile (I-P2)

This record defines the ICS envelope settings being used. The fields are described in Table 68.

Table 68. ICS (I envelope) profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **I**
ICSKEY	12–19	8	Char	Profile name
ICS01	20–21	2	Char	Subelement separator
ICS02	22–25	4	Char	Control standards ID
ICS03	26–30	5	Char	Control version number
ICS04	31–32	2	Char	Sender ID qualifier
ICS05	33–47	15	Char	Information sender ID
ICS06	48–49	2	Char	Receiver ID qualifier

Table 68. ICS (I envelope) profile (continued)

Tag	Position	Length	Type	Description
ICS07	50–64	15	Char	Information receiver ID
ICS08	65–70	6	Char	Interchange date
ICS09	71–74	4	Char	Interchange time
ICS10	75–83	9	Char	Interchange control number
ICE01	84–89	6	Char	Number of groups
ICE02	90–98	9	Char	Interchange control number
GS01	99–100	2	Char	Functional group ID
GS02	101–115	15	Char	Application sender code
GS03	116–130	15	Char	Application receiver code
GS04	131–136	6	Char	Functional group date
GS05	137–144	8	Char	Functional group time
GS06	145–153	9	Char	Functional group control number
GS07	154–155	2	Char	Responsible agency code
GS08	156–167	12	Char	Version/release/industry ID
GE01	168–173	6	Char	Number of included sets
GE02	174–182	9	Char	Functional group control number
ST01	183–185	3	Char	Transaction set ID
ST02	186–194	9	Char	Transaction set control number
SE01	195–204	10	Char	Number of included segments
SE02	205–213	9	Char	Transaction set control number
DESCRIPT	214–243	30	Char	Description

UN/TDI (T envelope) profile (T-P2)

This record defines the UN/TDI envelope settings being used. The fields are described in Table 69.

Table 69. UN/TDI (T envelope) profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **T**
TDIKEY	12–19	8	Char	Profile name
STX01	20–23	4	Char	Syntax identifier
STX02	24	1	Char	Syntax version number
STX03	25–38	14	Char	Sender code
STX04	39–73	35	Char	Sender name
STX05	74–87	14	Char	Recipient code
STX06	88–122	35	Char	Recipient name
STX10	123–136	14	Char	Recipient's reference/password

Table 69. UN/TDI (T envelope) profile (continued)

Tag	Position	Length	Type	Description
STX11	137–150	14	Char	Application reference
STX12	151	1	Char	Transmission priority code
MHD03	152	1	Char	Message version
DESCRIPT	153–182	30	Char	Description

UCS (U envelope) profile (U-P2)

This record defines the UCS envelope settings being used. The fields are described in Table 70.

Table 70. UCS (U envelope) profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **U**
UCSKEY	12–19	8	Char	Profile name
BG01	20–29	10	Char	Communication ID
BG02	30–39	10	Char	Communication password
BG03	40–54	15	Char	Application sender ID
BG04	55–79	15	Char	Application receiver ID
GS01	80–91	2	Char	Functional group ID
GS02	92–106	15	Char	Application sender's code
GS03	107–121	15	Char	Application receiver's code
GS07	122–123	2	Char	Responsible agency code
GS08	124–135	12	Char	Version/release/industry ID
ST03	136–170	35	Char	Implementation convention reference
DESCRIPT	171–200	30	Char	Description

X12 (X envelope) profile (X-P2)

This record defines the X12 envelope settings being used. The fields are described in Table 71.

Table 71. X12 (X envelope) profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **X**
X12KEY	12–19	8	Char	Profile name
ISA01	20–21	2	Char	Authorization information qualifier
ISA02	22–31	10	Char	Authorization information
ISA03	32–33	2	Char	Security information qualifier
ISA04	34–43	10	Char	Security information
ISA05	44–45	2	Char	Interchange sender ID qualifier
ISA06	46–60	15	Char	Interchange sender ID
ISA07	61–62	2	Char	Interchange receiver ID qualifier
ISA08	63–77	15	Char	Interchange receiver ID
ISA11	88	1	Char	Interchange standard ID
ISA12	89–93	5	Char	Interchange version ID
ISA14	94	1	Char	Acknowledgment requested
ISA16	95	1	Char	Subelement separator

Table 71. X12 (X envelope) profile (continued)

Tag	Position	Length	Type	Description
GS01	96–97	2	Char	Functional group ID
GS02	98–112	15	Char	Application sender's code
GS03	113–127	15	Char	Application receiver's code
GS07	128–129	2	Char	Responsible agency code
GS08	130–142	12	Char	Version/release/industry ID
ST03	143–177	35	Char	Implementation convention reference
DESCRIPT	178–207	30	Char	Description

Continuous receive profile (CONTRECV-P2 for CICS only)

This record defines the continuous receive settings being used. The fields are described in Table 72.

Table 72. Continuous receive profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **CONTRECV**
CONTRCV	12–27	16	Char	Profile name
ACTIVE	28	1	Char	Profile active
REQID	29–44	16	Char	Mailbox (requestor) ID
TPNICKN	45–60	16	Char	Trading Partner nickname
MSGCLS	61–68	8	Char	Network message user class
TRANSLATE	69	1	Char	Deenvelope and translate
RAWDATA	70	1	Char	Generate raw data
PRINTNM	71–78	8	Char	Print file name
PRINTTYP	79–80	2	Char	Print file type
EXCPNM	81–88	8	Char	Exception file name
EXCPTP	89–90	2	Char	Exception file type
ADDLRECS	91–95	5	Char	Additional records
DENVONLY	96	1	Char	Deenvelope only
DELAYFA	97	1	Char	Delay functional acknowledgment enveloping
FATSQ	98–105	8	Char	Enveloped functional acknowledgment TS queue name
RESPNM	106–113	8	Char	Name of response application TD queue
RESPTP	114–115	2	Char	Type of response
USERFLD	116–131	16	Char	User field passed to response
APPLID	132–139	8	Char	Application ID

Table 72. Continuous receive profile (continued)

Tag	Position	Length	Type	Description
NLSID	140–145	6	Char	National language ID
SYNCPTS	146	1	Char	Enable syncpoints
DUPENVL	147	1	Char	Enable duplicate envelopes
NETAKONLY	148	1	Char	Process network acknowledgments
PURGEINT	149–152	4	Dec	Transaction purge interval
DESCRIPT	153–182	30	Char	Description
SAPUPDT	183	1	Char	SAP update
PAGE	184	1	Char	Pageable translation active flag

CICS performance profile (SYSPROF-P2 for CICS only)

This record defines the CICS performance settings being used. The fields are described in Table 73.

Table 73. CICS performance profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **SYSPROF**
SYSID	12–19	8	Char	System ID
PERSACTV	20	1	Char	Persistent environment active
PERSIZE	21–24	4	Dec	Persistent environment size
PERSTHDS	25–26	2	Dec	Number persistent environment threads
DESCRIPT	27–56	30	Char	Description

WebSphere MQ queue profile (WebSphere MQ-P2)

This record defines the WebSphere MQ queue settings being used. The fields are described in Table 74.

Table 74. WebSphere MQ queue profile

Tag	Position	Length	Type	Description
PROFID	4–11	8	Char	Profile ID = **MQSERIES**
QUEUEID	12–19	8	Char	Logical name associated with the queue name
REALNAME	20–67	48	Char	Actual WebSphere MQ queue name
MGRNAME	68–115	48	Char	Override queue manager name
READFLAG	116	1	Char	Destructive read flag indicator
MQSYNCFLAG	117	1	Char	Syncpoint control
MAXMSGLN	118–125	8	Char	Maximum message length

Table 74. WebSphere MQ queue profile (continued)

Tag	Position	Length	Type	Description
DESCRIPT	126–155	30	Char	Description

Transformation Table—Data References

This record defines the Data References defined for the Transformation Table. The fields are described in Table 75.

Table 75. Transformation Table Data References

Tag	Position	Length	Type	Description
APPLICID	0-7	8	Char	Application ID
FILENAME	8-15	8	Char	Logical file or DD name
SVCFILENAME	16-23	8	Char	Service profile
DESCRIPT	24-53	30	Char	Description
FILEREF	54-308	255	Char	Real file name
MAXRECL	309-312	4	Char	Largest logical record
INITIALDISP	313	1	Char	Initial disposition
ERRORDISP	314	1	Char	Disposition on error
EXITDISP	315	1	Char	Disposition if okay
RECDELIM	316-317	2	Char	Record delimiter code
UNICODE	318-333	16	Char	Coded character set name
ACCGROUP	334-363	30	Char	Access group
FOLDER	364	64	Char	Folder name

Transformation Table—Security Roles

This record defines the Security Roles defined in the Transformation Table. The fields are described in Table 76.

Table 76. Transformation Table Security Roles

Tag	Position	Length	Type	Description
N2				
ROLE	0-29	30	Char	Role name
DESCRIPT	30-79	80	Char	Description
FOLDER	110	64	Char	Folder
N5				
ROLE	0-29	30	Char	Role name
ROLEREF	30	80	Char	Role reference
N6				
ROLE	0-29	30	Char	Role name
OBJECTID	30-35	6	Char	Object ID

Table 76. Transformation Table Security Roles (continued)

Tag	Position	Length	Type	Description
ACCESS	36	6	Char	Access value

Transformation Table—Security UserIDs

This record defines the Security UserIDs defined in the Transformation Table. The fields are described in Table 77.

Table 77. Transformation Table Security Roles

Tag	Position	Length	Type	Description
N3				
USERID	0-16	17	Char	User ID
ALLGRPS	17	1	Char	All groups
DESCRIPT	18-97	80	Char	Description
FOLDER	98	64	Char	
N7				
USERID	0-16	17	Char	User ID
ROLEREF	17	30	Char	Role reference
N8				
USERID	0-16	17	Char	User ID
OBJECTID	17-22	6	Char	Object ID
ACCESS	23	6	Char	Access value

Transformation Table—Security Groups (N9)

This record defines the Security Groups defined in the Transformation Table. The fields are described in Table 78.

Table 78. Transformation Table Security Roles

Tag	Position	Length	Type	Description
N4				
ACCGROUP	0	30	Char	Access group
DESCRIPT	30	80	Char	Description
FOLDER	110	64	Char	Folder
N9				
USERID	0	17	Char	User ID
ACCGROUP	17	30	Char	Access group

Transformation Table—Command Dictionary

This record defines the Command Dictionary defined in the Transformation Table. The fields are described in Table 79 on page 284.

Table 79. Transformation Table Security Roles

Tag	Position	Length	Type	Description
DICTIONARYNAME	0-29	30	Char	Dictionary name
DESCRIPT	30-59	30	Char	Description
PRIVATE	60	1	Char	Private
OWNERUID	61-77	17	Char	Owner userid
ACCGROUP	78-107	30	Char	Access group
FOLDER	108-171	64	Char	Folder
CMTTXT	172	2048	Char	Comment text

Transformation Table—Commands Table

This record defines the Commands Table defined in the Transformation Table. The fields are described in Table 80.

Table 80. Transformation Table Security Roles

Tag	Position	Length	Type	Description
DICTIONARYNAME	0-29	30	Char	Dictionary name
CMDNAME	30	30	Char	Command name
DESCRIPT	60-89	30	Char	Description
ACCGROUP	90-119	30	Char	Access group
FOLDER	120-183	64	Char	Folder
COMMAND	184-8375	8192	Char	Command
CMTTXT	8376	2048	Char	Comment text

Transformation Table—Commands Reference Table

This record defines the Commands Reference Table defined in the Transformation Table. The fields are described in Table 81.

Table 81. Transformation Table Security Roles

Tag	Position	Length	Type	Description
DICTIONARYNAME	0-29	30	Char	Dictionary name
CMDNAME	30-59	30	Char	Command name
FILENAME	60-67	8	Char	File name
FILEREF	68	255	Char	File path

Chapter 5. Utility records format

WebSphere Data Interchange needs a description of the data format for each business application that generates data that will be translated, or uses data that has been translated. An application's data must be described to WebSphere Data Interchange so that it can be used as a source or target for translation.

WebSphere Data Interchange can accept two types of data record formats: raw data records and control and data records, as described later in this section.

With Raw data, each record in raw data format identifies itself by containing a unique record identifier (a record ID). Raw data can be either comma-separated or fixed-position. For fixed-position data, the identifier starts in the same position and extends for the same length in each record. In comma-separated data, each value is separated by a comma. Comma-separated data is used only with data transformation maps. Raw data that is fixed-position can be used in data transformation maps, send maps and receive maps. The Record ID is actually a field in the record. As long as records contain identifiable record IDs, you can use application data without modification.

Control and Data (C and D) records are used when no record identifier exists clearly specifying the type of information contained in the record.

Note: For Data Transformation processing, the data format definition used during mapping must also be defined as C and D or Raw Data application record type on the Record ID Info tab.

This section describes the format of the WebSphere Data Interchange Utility records, which are:

Control (C) records

The C record format is used for both send and receive transactions. All fields are left-justified and case-sensitive. The expanded fields (position 67 and greater) are optional. Use these fields to provide the override values you want the translator or enveloper to use.

The advantages of using C and D records are:

- You can use multiple formats within a single file.
- You can provide overrides for fields within service segments (ISA, GS, UNB, UNH, and so on).

The C record must identify the trading partner and the Data Format. With send/receiving map processing the internal trading partner ID field is used for Send Processing. Data Transformation processing currently requires the application and EDI trading partner fields.

Control record format

Table 82 describes the record layout for translating C record data to EDI standard format.

Table 82. Control (C) record for translating to standard format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **C**
INTPID	2–36	35	Char	Internal trading partner ID
FORMATID	37–52	16	Char	Data format ID
TRANRC	53–56	4	Integer	Translator return code
TRANXRC	57–60	4	Integer	Translator extended return code
UTILRC	61–64	4	Integer	WebSphere Data Interchange Utility return code
TESTIND	65	1	Char	Usage indicator
MUWIND	66	1	Char	Multiple D records indicator
XPANDED	67	1	Char	For send only. Expanded control block indicator
ITYPE	68	1	Char	For send only. Return information record indicator
ETYPE	69	1	Char	For send only. Return envelope header record indicator
GTYPE	70	1	Char	For send only. Return group header record indicator
TTYPE	71	1	Char	For send only. Return transaction set header record indicator
QTYPE	72	1	Char	For send only. Return queuing totals record indicator
ISID	73–107	35	Char	Interchange sender ID
IRID	108–142	35	Char	Interchange receiver ID
IVERREL	143–147	5	Char	Interchange version and release number
ISPW	148–161	14	Char	Interchange password
IAPREF	162–175	14	Char	Interchange application reference
GSID	176–210	35	Char	Group application sender ID
GRID	211–245	35	Char	Group application receiver ID
GVER	246–257	12	Char	Group version
GREL	258–269	12	Char	Group release number
GAPW	270–283	14	Char	Group password

Table 82. Control (C) record for translating to standard format (continued)

Label	Position	Length	Type	Description
TVER	284–289	6	Char	Transaction version
TREL	290–295	6	Char	Transaction release number
HOLDFLAG	296	1	Char	Held status indicator
BNDLFLAG	297	1	Char	Bundle indicator
ROUTCODE	298–300	3	Char	Generic routing code
ISYNTAXID	301–304	4	Char	Interchange syntax ID for **E** and **T** envelopes
ISYNTAXVER	305	1	Char	Interchange syntax version for **E** and **T** envelopes
ISIDQUAL	306–309	4	Char	Interchange sender ID qualifier
ISENDNAME	310–323	14	Char	Interchange sender name for **T** envelope) Application sender code for **U** envelope
IREVROUT	324–337	14	Char	Interchange reverse routing for **E** envelope
IRIDQUAL	338–341	4	Char	Interchange receiver ID qualifier
IRECVNAME	342–355	14	Char	Interchange receiver name for **T** envelope) Application receiver code for **U** envelope
IROUTEADDR	356–369	14	Char	Interchange routing address for **E** envelope
ISTDID	370–373	4	Char	Interchange standard ID for **I** and **X** envelopes
IPRIOR	374	1	Char	Interchange priority for **E** and **T** envelopes
ICOMMAGREE	375–409	35	Char	Interchange communication agreement for **E** envelope
GSIDQ	410–413	4	Char	Group application sender ID qualifier
GRIDQ	414–417	4	Char	Group application receiver ID qualifier
GRESPAG	418–419	2	Char	Group responsible agency code
DUPTRANS	420	1	Char	For receive only. Duplicate transaction indicator
FORCEC	421	1	Char	For send only. Force C record format on send translation
APPLTPID	422–437	16	Char	Application trading partner
EDITPID	438–453	16	Char	EDI trading partner
MAPID	454–469	16	Char	Name of map to be used for translation
TRXSTATUS	470	1	Char	C Flag F (failed) or Y (successful)
RSRVD1	471–512	42	Char	Reserved

Table 82. Control (C) record for translating to standard format (continued)

Label	Position	Length	Type	Description
CUSERDATA	513–768	256	Char	User data area
RSRVD2	769–1024	256	Char	Reserved

Control record label descriptions

RECID
The transaction set control record with a value of **C**.

INTPID
The internal trading partner ID used to define the map.

FORMATID
The data format ID used to describe the application data.

TRANRC
The translation status of the transaction. The code is in binary format.

TRANXRC
Defines the translation status of the transaction. The code is in binary format.

UTILRC
For send translation, indicates whether a transaction is written to the exception file in the UTILRC field. The WebSphere Data Interchange Utility returns the binary value **8** in UTILRC if it writes the transaction to the exception file.

For receive translation, UTILRC is always a binary **0**.

TESTIND
Indicates the usage/rule used for the transaction. Valid values are:

I	Information transaction. Use an information usage/rule. If an information usage/rule is not found, use a production usage/rule. Even when a production usage/rule is used, the transaction is flagged as an information transaction.
P	Production transaction. Use only a production usage/rule (default).
T	Test transaction. Use a test usage/rule. If a test usage/rule is not found, use a production usage/rule. Even when a production usage/rule is used, the transaction is flagged as a test transaction.
U	The translator should determine if the transaction is test, information, or production based on the usage/rule found. If a test usage/rule is found, the transaction is a test transaction, and value of **T** is returned. If an information usage/rule is found, the transaction is an information transaction, and a value of **I** is returned. If only a production usage/rule is found, the transaction is a production transaction, and a value of **P** is returned.

MUWIND

Indicates whether the transaction contains a single D record or multiple D records are involved in the transaction. Valid values are:

Y	Multiple D records are involved in the transaction.
N	A single D record is involved in the translation.

XPANDED

For send translation only. Indicates the version of the C record format being used (which fields are contained in the record). Receive translation always writes all fields through column 1024. Valid values are:

This value:	Specifies the C record contains fields through:
Y	Column 300
N	Column 66
1	Column 453
2	Column 1024

ITYPE

Indicates whether an information record is written to the exception/tracking file. Generally used for send translation. On receive translation, the field contains the same values as shown for “DUPTRANS” on page 296. Valid values are:

Y	Writes an information record to the exception/tracking file for each translated transaction. For more information, see “Information (I) records” on page 303.
N	Does not write an information record.

ETYPE

For send translation only. Indicates whether an interchange header image is written to the exception/tracking file. Valid values are:

Y	Writes an image of the interchange header segment to the exception/tracking file for each interchange processed. For more information, see “Interchange header (E) records” on page 304.
N	Does not write an image of the interchange header segment.

GTYPE

For send translation only. Indicates whether a group header image is written to the exception/tracking file. Valid values are:

Y	Writes an image of the group header segment to the exception/tracking file for each group processed. For more information, see “Group header (G) records” on page 305.
N	Does not write an image of the group header segment.

TTYPE

For send translation only. Indicates whether a transaction header image is written to the exception/tracking file. Valid values are:

Y	Writes an image of the transaction header segment to the exception/tracking file for each transaction processed. For more information, see “Transaction set header (T) records” on page 305.
N	Does not write an image of the transaction header segment.

QTYPE

For send translation only. Indicates whether a record containing totals relative to an interchange is written to the exception/tracking file. Valid values are:

Y	Writes information relative to totals in an interchange to the exception/tracking file for each interchange processed. For more information see “Queuing totals (Q) records ” on page 305.
N	Does not write interchange totals information.

ISID

Overrides the ID provided in the envelope profile member or the trading partner profile member. It maps to a type **IS** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB03	E
ICS05	I
STX03 or STX04	T
BG03	U
ISA06	X

IRID

Overrides the ID provided in the trading partner profile or envelope profile member. It maps to a type **IR** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB06	E
ICS07	I
STX05 or STX06	T
BG04	U
ISA08	X

IVERREL

Overrides the version and release provided in the envelope profile member. It maps to types **VR** and **LV** standard data elements. WebSphere Data Interchange provides the equivalent data on receive translation.

ISPW

Overrides the password that the trading partner profile member provides. It maps to a type **PW** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB12	E
STX10	T
ISA04	X

IAPREF

Overrides the application reference ID provided in the envelope profile member. WebSphere Data Interchange uses this reference as the message user class for EDIFACT and UN/TDI. This reference points to a type **AP** standard data element. As distributed by WebSphere Data Interchange , no fields contain the **AP** data type. However, the following fields are frequently customized to have a data type of **AP**. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB14	E
STX11	T

GSID

Overrides the data format ID provided in the map. If the map provides the name of an envelope profile member, this entry overrides the sender ID that the envelope profile member provides. The group application sender ID maps to a type **AS** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNG02	E
GS02	X
GS02	I
GS02	U

GRID

Overrides the trading partner application name provided in the map. If the map provides the name of an envelope profile member, this entry overrides the receiver ID that the

envelope profile member provides. The group application ID maps to a type **AR** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNG04	E
GS03	X
GS03	I
GS03	U

GVER

Overrides the group version provided in the envelope profile member. The value in the group version maps to a type **VR** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation.

GREL

Overrides the group release number provided in the envelope profile member. The group release number maps to type **LV** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation.

GAPW

Overrides the group password provided in the trading partner usages/rules. This usage/rule overrides the password in the envelope profile member. The group password maps to a type **PW** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation. Valid value is:

This value:	Overrides this envelope type:
UNG13	E

TVER

Overrides the transaction version provided in the envelope profile member. The transaction version maps to a type **VR** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation.

TREL

Overrides the transaction release number provided in the envelope profile member. The transaction release number maps to a type **LV** standard data element. WebSphere Data Interchange provides the equivalent data on receive translation.

HOLDFLAG

Specifies if the transaction is in hold status. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

Y	Places the transaction in held status.

N	Does not hold the transaction. Transactions that are not on hold and do not have a date specified for enveloping are available immediately for enveloping and sending.

BNDLFLAG

Specifies if the transaction starts a bundle. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

Y	Starts a group of related transactions (a bundle). All transactions that follow are part of the bundle until the translator encounters another transaction with this field set to **Y** or **N**.
N	Ends the bundle without starting a new one.
(blank)	Continues processing as usual. The translator ends a bundle automatically if the data forces the start of a new group or interchange envelope. Any action that changes the transaction or store status of one member of the bundle (such as envelope, send, or hold) is applied to all members of the bundle.

Note: If the current trading partner is not using functional groups (value of **N** in the `Functional group` field of the trading partner profile), WebSphere Data Interchange ignores the changes in the data. A new group is created but does not end the bundle.

ROUTCODE

For send translation only. A three-character generic routing code provided by the application and used by WebSphere Data Interchange to select a generic send usage/rule. A blank specifies a default generic send usage/rule. WebSphere Data Interchange provides the equivalent data on receive translation.

ISYNTAXID

Overrides the interchange syntax ID provided in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB01	E
STX01	T

ISYNTAXVER

Overrides the interchange syntax version provided in the envelope profile member. Valid values are:

This value:	Overrides this envelope type:
UNB02	E
STX02	T

ISIDQ

Overrides the interchange sender ID qualifier provided in the envelope profile member or trading partner profile member. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB04	E
ICS04	I
ISA05	X

ISENDNAME

Overrides the interchange sender name provided in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
STX04	T
UCS03	U (If not IS data type)

IREVROUT

Overrides the interchange reverse routing provided in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. This entry must be left-justified. Valid values are:

This value:	Overrides this envelope type:
UNB05	E

IRIDQ

Overrides the interchange receiver ID qualifier provided in the trading partner profile or envelope profile member. Valid values are:

This value:	Overrides this envelope type:
UNB06	E
ICS06	I
ISA07	X

IRECVNAME

Overrides the interchange receiver name provided in the envelope profile member. This entry must be left-justified. Valid values are:

This value:	Overrides this envelope type:
STX06	T

This value:	Overrides this envelope type:
UCS04	U (If not **IR** data type)

IROUTADDR

Overrides the interchange routing address provided in the envelope profile member. This entry must be left-justified. Valid value is:

This value:	Overrides this envelope type:
UNB08	E

ISTDID

Overrides the interchange standard ID provided in the envelope profile member. This entry must be left-justified. Valid values are:

This value:	Overrides this envelope type:
ICS02	I
ISA11	X

IPRIOR

Overrides the interchange processing priority defined in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB15	E
STX12	T

ICOMMAGREE

Overrides the interchange communication agreement provided in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNB17	E

GSIDQ

Overrides the group sender ID qualifier provided in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. This entry must be left-justified. Valid values are:

This value:	Overrides this envelope type:
UNG03	E

GRIDQ

Overrides the group receiver ID qualifier provided in the envelope profile member. This entry must be left-justified. Valid values are:

This value:	Overrides this envelope type:
UNG05	E

GRESPAG

Overrides the group responsible agency code and controlling agency provided in the envelope profile member. WebSphere Data Interchange provides the equivalent data on receive translation. Valid values are:

This value:	Overrides this envelope type:
UNG09	E
GS07	I
GS07	U
GS07	X

DUPTRANS

For receive translation only. Specifies if a transaction is part of a duplicate envelope. Valid values are:

Y	Part of a duplicate envelope
N	Not part of a duplicate envelope

FORCEC

For send translation only. Forces a C record to be written. Valid values are:

Y	Always writes a C record
N or (blank)	Only writes a C record when an error occurs (default)

APPLTPID

The trading partner nickname of the application trading partner as defined in the trading partner profile.

EDITPID

The trading partner nickname of the EDI trading partner as defined in the trading partner profile.

RSVD1
Reserved for WebSphere Data Interchange .

CUSERDATA
The value in this field is copied to the translator control block (TRCB) where it can be modified by user exits. Before any C record is output by WebSphere Data Interchange , the value in this field is copied from the equivalent field in the TCRB. On receive translation, the value in the TRCB can be set using the WebSphere Data Interchange reserved variable DICUSERDATA. WebSphere Data Interchange does not use this field.

RSVD2
Reserved for WebSphere Data Interchange .

Data Transformation Node Names/AMM Structure for C record
The Data Format Parser creates the FFCREC from the C record information. A customer written parser can also create an FFCREC Abstract Message Model node with any combination of the name and value nodes to enable the Data Format Deenveloper or Enveloper to process or create a customer defined C record layout.

```
...body (N)
......FFCREC (N)
.........INTPID (NV) = (char) 'MMTEPO1'
.........FORMATID (NV) = (char) 'MMTEPO1CDXX'
.........TESTIND (NV) = (char) 'N'
.........MUWIND (NV) = (char) 'Y'
.........XPANDED (NV) = (char) '1'
.........ITYPE (NV) = (char) 'Y'
.........ETYPE (NV) = (char) ' '
.........GTYPE (NV) = (char) ' '
.........TTYPE (NV) = (char) ' '
.........QTYPE (NV) = (char) ' '
.........ISID (NV) = (char) 'MMTEPO1'
.........IRID (NV) = (char) 'MMTEPO1'
.........IVERREL (NV) = (char) ' '
.........ISPW (NV) = (char) ' '
.........IAPREF (NV) = (char) ' '
.........GSID (NV) = (char) ' '
.........GRID (NV) = (char) ' '
.........GVER (NV) = (char) ' '
.........GREL (NV) = (char) ' '
.........GAPW (NV) = (char) ' '
.........TVER (NV) = (char) ' '
.........TREL (NV) = (char) ' '
.........HOLDFLAG (NV) = (char) ' '
.........BNDLFLAG (NV) = (char) ' '
.........ROUTCODE (NV) = (char) ' '
.........ISYNTAXID (NV) = (char) 'UNA'
.........ISYNTAXVER (NV) = (char) ' '
.........ISIDQUAL (NV) = (char) 'ZZ'
.........ISENDNAME (NV) = (char) ' '
.........IREVROUT (NV) = (char) ' '
.........IRIDQUAL (NV) = (char) 'ZZ'
.........IRECVNAME (NV) = (char) ' '
```

```
.........IROUTEADDR (NV) = (char) ' '
.........ISTDID (NV) = (char) ' '
.........IPRIOR (NV) = (char) ' '
.........ICOMMAGREE (NV) = (char) ' '
.........GSIDQ (NV) = (char) ' '
.........GRIDQ (NV) = (char) ' '
.........GRESPAG (NV) = (char) ' '
.........DUPTRANS (NV) = (char) ' '
.........FORCEC (NV) = (char) ' '
.........APPLTPID (NV) = (char) 'ANY'
.........EDITPID (NV) = (char) 'ANY'
.........MAPID (NV) = (char) ' '
.........CUSERDATA (NV) = (char) '(Value too large for buffer - 256 bytes)'
```

Data (D) records

There are two formats for data (D) records. One format is used when all the data for a transaction is provided by a single structure. The other format is used when data for a transaction is provided by multiple structures. The format of D records is described in the following tables.

For Fixed-to-Fixed mapping, when there is no target data format, the STRUCTNAME is the segment ID value from the EDI standard transaction definition, and D records are always output in the format described for single structures on 299.

When translating D records, use the name in the `ID field`, and populate ISA and GS segments with the values passed in the C record.

Notes:

1. The largest record your application or the translator can handle is 32000 bytes, which is the largest LRECL for QSAM files. During translate-to-application processing, WebSphere Data Interchange divides any records that are larger than 32000 bytes into one or more X records, and uses a D record as the last record of the structure. For example, a structure that is 80000 bytes in length requires two X records (64000 bytes) followed by one D record (16000 bytes). During translate-to-EDI-standard processing, the program that creates the application file must create X and D records for structures exceeding 32000 bytes in length. Define the application file as variable-blocked (VB) if the application file will contain X records.
2. X records are not required for WebSphere Data Interchange Data Transformation. The keyword, XRECS, is used with the PERFORM command to indicate that no X records are generated for inbound processing with C and D data format and application data is the target message.

Data record format (single structure)

Table 83. Data record format: single structure

Label	Position	Length	Type	Description
RECID	1	1	Char	Record identifier

Table 83. Data record format: single structure (continued)

Label	Position	Length	Type	Description
DATARCD	2–32750	32749	Char	Application transaction data

Data record label descriptions (single structure)

The following are descriptions of the labels for the data record when data structures are passed together.

RECID

Specifies if a transaction exceeds 32000 bytes. Valid values are:

D

An entire transaction, or the last record of a transaction that exceeds 32000 bytes

X

The first and middle records of a transaction that exceed 32000 bytes

DATARCD

Specifies application transaction data.

Data record format (multiple structures)

Table 84. Data record format: multiple structures

Label	Position	Length	Type	Description
RECID	1	1	Char	Record identifier
STRUCNAM	2–17	16	Char	Structure name
DATARCD	18–32750	32733	Char	Application transaction data

Data record label descriptions (multiple structures)

The following are descriptions of the labels for the data record when data structures are passed separately.

RECID

Specifies if a transaction exceeds 32000 bytes. Valid values are:

D

An entire transaction, or the last record of a transaction that exceeds 32000 bytes

X

The first and middle records of a transaction that exceed 32000 bytes

STRUCNAME
The structure name defined in the data format for this transaction.

DATARCD
Specifies application transaction data.

End transaction and interchange (Z) records

The Z record is optional and is used in the z/OS environment when the size of an interchange is being controlled by the application, or in the CICS environment when recoverable resources are being used. The Z record marks the end of:

- The application data for a transaction. If a Z record is not present, WebSphere Data Interchange can only detect the end of a transaction from reading the next transaction's C record.

  If your input data is in recoverable intrapartition TD queues, using the Z record keeps the reading of the C record for the next transaction from being part of the syncpoint interval for the current transaction. The Z record provides a formal end for a transaction rather than the implied transaction end achieved when the next C record is read.
- The application data for a transaction and indicates that the current transaction is the last transaction for an interchange when a Z record with the ENDINTERCH field set to 1 (Z1) is present. If a Z1 record is not present, WebSphere Data Interchange can only detect the end of an interchange from reading the next transaction's C record.

  If your input data is in recoverable intrapartition TD queues, using a Z1 record keeps the reading of the C record for the next transaction from being part of the syncpoint interval for the current interchange. The Z1 record provides a formal end for an interchange rather than the implied end achieved when the next C record is read. The Z1 record can be used to artificially limit the size of an interchange. A Z1 record causes the current interchange to be completed. A new interchange starts with the next transaction.

Z and Z1 records are only necessary when using the C and D record format, and when the conditions are true. Otherwise, using Z and Z1 records is optional. The conditions for use are:

- The WebSphere Data Interchange Utility is being executed in the CICS environment.
- Application data is given to the WebSphere Data Interchange Utility in recoverable intrapartition TD queues.
- WebSphere Data Interchange issues **CICS SYNCPOINT** commands.

Z record format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **Z**
ENDINTERCH	2	1	Char	End of interchange indicator

Z record label descriptions

RECID
A value of **Z** identifies this record as a transaction or interchange terminator.

ENDINTERCH
A value of **1** indicates the preceding transaction is the last in the current interchange. The next C record begins a new transaction and new interchange.

Raw data records

Table 85 describes the format of raw data records. The data format must indicate the position, length, and type of record ID field within the data. The data format can also provide the field in the application data that contains the internal trading partner ID value. For TRANSLATE TO STANDARD processing, the data format ID is provided by the `RAWFMTID` keyword on the PERFORM command. During TRANSLATE TO APPLICATION processing, the translator automatically supplies the record ID values. The internal trading partner ID value is taken from the trading partner receive usage/rule when raw data is requested.

WebSphere Data Interchange can create raw data output during TRANSLATE TO STANDARD processing with Fixed-to-Fixed mapping. Record ID values and the internal trading partner ID default values are used but might be overridden during mapping.

Raw data record format

Table 85. Raw data record format

Label	Position	Length	Type	Description
DATARCD	1–32750	32750	Char	Application transaction data

Optional records

The following optional record types are available:
- “Information (I) records” on page 303
- “Interchange header (E) records” on page 304
- “Group header (G) records” on page 305
- “Transaction set header (T) records” on page 305
- “Queuing totals (Q) records ” on page 305
- “File (F) records” on page 306

You can request optional records using the `TYPE` fields in the control record (“Control (C) records” on page 285), on the Additional Records panel, or by using the OPTRECS keyword (“OPTRECS” on page 179) in one of the following commands:

```
DEENVELOPE
DEENVELOPE AND TRANSLATE
ENVELOPE
ENVELOPE AND SEND
```

```
RECEIVE AND DEENVELOPE
RECEIVE AND TRANSLATE
REENVELOPE
REENVELOPE AND SEND
RETRANSLATE TO APPLICATION
TRANSLATE AND ENVELOPE
TRANSLATE AND SEND
TRANSLATE TO APPLICATION
TRANSLATE TO STANDARD
```

Different optional records are available for different commands. The commands and the record types they support are defined in Table 86.

Table 86. Optional records supported by command

	Record type					
Command	**E**	**F**	**G**	**I**	**Q**	**T**
DEENVELOPE DEENVELOPE AND SEND	X	X	X	X	X	X
ENVELOPE ENVELOPE AND SEND	X		X	X	X	X
RECEIVE AND DEENVELOPE RECEIVE AND TRANSLATE	X		X	X	X	X
REENVELOPE REENVELOPE AND SEND	X		X	X	X	X
RETRANSLATE TO APPLICATION						
TRANSLATE AND ENVELOPE	X		X	X	X	X
TRANSLATE AND SEND	X		X	X	X	X
TRANSLATE TO APPLICATION				X		X
TRANSLATE TO STANDARD				X		

For TRANSLATE TO STANDARD operations, the WebSphere Data Interchange Utility writes the optional records to a file as follows:

- Writes the records to FFSTRAK if it exists.
- Writes the records to FFSEXCP if FFSTRAK does not exist.
- Writes no records if FFSTRAK does not exist and raw data (RAWFMTID keyword) is requested.

For TRANSLATE TO APPLICATION operations, the WebSphere Data Interchange Utility writes the optional records to a file as follows:

- Writes the records to the application output file if C and D records are requested.
- Writes the records to FFSEXCP if the raw data records are requested (RAWDATA keyword).

For DEENVELOPE operations, the WebSphere Data Interchange Utility writes the optional records to the FFSEXCP file.

Information (I) records

Table 87 describes the format and contents of the information (I) record. You can request an information record on the Optional Record Options panel, in the `ITYPE` field in the control record, or by using the `OPTRECS` keyword in a command. WebSphere Data Interchange returns one record for each transaction with the values that were present at the time the transaction was translated.

Table 87. Information record format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **I**
IHXCTL	2–15	14	Char	Interchange header control number
ISID	16–50	35	Char	Interchange sender ID
IRID	51–85	35	Char	Interchange receiver ID
IDATE	86–91	6	Char	Interchange date
ITIME	92–97	6	Char	Interchange time
IVERREL	98–102	5	Char	Interchange version/release
IGT	103–108	6	Char	For sending, the total number of groups in the interchange at the current time For receiving, the total number of groups processed so far in the interchange
ITT	109–114	6	Char	For sending, the total number of transactions in the interchange at the current time For receiving, the total number of transactions processed so far in the interchange
IST	115–124	10	Char	For sending, the total number of segments in the interchange at the current time For receiving, the total number of segments processed so far in the interchange
IBT	125–132	8	Char	For sending, the total number of bytes in the interchange at the current time For receiving, the total number of bytes processed so far in the interchange
ISPW	133–146	14	Char	Interchange password
IAPREF	147–160	14	Char	Interchange application reference
GHXCTL	161–174	14	Char	Control number of group header
GFGID	175–180	6	Char	Group functional group ID

Table 87. Information record format (continued)

Label	Position	Length	Type	Description
GSID	181–215	35	Char	Group application sender ID
GRID	216–250	35	Char	Group application receiver ID
GDATE	251–256	6	Char	Group date
GTIME	257–262	6	Char	Group time
GVER	263–274	12	Char	Group version
GREL	275–286	12	Char	Group release
GTT	287–292	6	Char	For sending, current group transaction total For receiving, total transactions in group
GAPW	293–306	14	Char	Group password
THXCTL	307–320	14	Char	Control number of transaction set header
TTC	321–326	6	Char	Transaction code
TVER	327–332	6	Char	Transaction version
TREL	333–338	6	Char	Transaction release
TST	339–348	10	Char	Transaction segment total
AC	349–383	35	Char	Application control field value
THANDLE	384–403	20	Char	Expanded value of the handle assigned to this transaction in the Document Store
TPNICKN	404–419	16	Char	Trading partner nickname
GDATE2	420–427	8	Char	Group date with century
RESERVED	428–483	56	Char	Reserved for future use

Interchange header (E) records

Table 88 shows the format and contents of the interchange (E) header record. You can request an interchange header record using the `ETYPE` field in the control record or the `OPTRECS` keyword in a command. WebSphere Data Interchange returns one record for each interchange that is created or received.

Table 88. Interchange header (E) record format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **E**
EDATA	2–256	255	Char	Standard interchange header image

Group header (G) records

Table 89 shows the format and contents of the group header (G) record. You can request a group header record using the GTYPE field in the control record or the OPTRECS keyword in a command. WebSphere Data Interchange returns one record for each group created or received.

Table 89. Group header (G) record format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **G**
GDATA	2–256	255	Char	Functional group header image

Transaction set header (T) records

Table 90 shows the format and contents of the transaction set header (T) record. You can request a transaction header record using the TTYPE field in the control record or the OPTRECS keyword in a command. WebSphere Data Interchange returns one record for each transaction created or received.

Table 90. Transaction set header (T) record format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **T**
TDATA	2–256	255	Char	Transaction set header image

Queuing totals (Q) records

Table 91 shows the format and contents of the queuing totals record. You can request a queuing totals record using the QTYPE field in the control record or the OPTRECS keyword in a command. WebSphere Data Interchange returns one record to the exception file each time an interchange is queued or deenveloped.

Table 91. Queueing totals (Q) record format

Label	Position	Length	Type	Description
RECID	1	1	Char	Record ID = **Q**.
QBT	2–9	8	Char	Number of bytes queued for the envelope.
QST	10–19	10	Char	Number of segments in the envelope.
QTT	20–25	6	Char	Number of transactions in the envelope.
QGT	26–31	6	Char	Number of groups in the envelope.
QDSNAME	32–87	56	Char	Physical data set name from which transactions were read or to which transactions were written. The name is terminated with a NULL character (X'00').

File (F) records

Table 92 shows the format and contents of the file (F) record. This finds the real file of the applications file.

Table 92. File (F) record layout

Label	Length	Type	Description
RECID	1	Char	Record ID = **F**.
FTYPE	1	Char	Type of data written. E=EDI D=ADF X=XML C=CSV
FLNAME	8	Char	Logical filename from **PERFORM** command
FPNAME	AIX/Windows - 272 z/OS - 56	Char	Physical file name
FRCVID	35	Char	Receiver ID from EDI header
FRCVQUAL	4	Char	Receiver qualifier from EDI header
FSNDID	35	Char	Sender ID from EDI header
FSNDQUAL	4	Char	Sender qualifier from EDI header
FNICKNAME	16	Char	Trading partner nickname from trading partner profile
FRECCNT	15	Char	Record count

Data Transformation C and D Support

The WebSphere Data Interchange Utility detects C and D data format processing when application data is the source and when there is no dictionary or document keyword specified on the PERFORM TRANSFORM command. Note the data format definition used during mapping must also be defined as C and D application record type on the Record ID Info tab. The C record FORMATID field identifies the document to be used and will be extracted and used during the translation process. The mapping special variable DICUserData can be used to map information from the source document to the C record field CUSERDATA when application data is the target document. This field can contain any data up to 256 bytes long.

Data Transformation processing does not currently use the internal trading partner ID; therefore the application and EDI trading partner fields will be used and the customer will need to use the fully expanded C record format which is 1024 bytes. The Abstract Message Model (AMM) node names will follow the names described in Chapter 5, “Utility records format,” on page 285. The name/value nodes will go under a name node FFCREC. The RECID field will NOT be present in the AMM since it is already known

that this is a 'C' record. Any C record value that identifies an override value will set the appropriate attribute. For example, the C record override ISID will set the attribute IchgSndrId as though a SetProperty command were issued within the mapping.

The following C record fields are not currently supported with data transformation processing:

```
INTPID 2-36 35 (Char Internal trading partner ID)
HOLDFLAG 296-296 1 (Char Held status indicator)
BNDLFLAG 297-297  1 (Char Bundle indicator)
ROUTCODE 298-300 3 (Char Generic routing code)
```

Management reporting

The management reporting data extracts are formatted as sequential files containing fixed-length (1024 bytes) records. All data extracts are written to the EDIQUERY file. Since other commands use the EDIQUERY file, you should define the EDIQUERY file as variable block 32756 rather than fixed block 1024 to accommodate commands that output larger records, such as the image records created by the ENVELOPE DATA EXTRACT and TRANSACTION DATA EXTRACT commands.

The output is tabular with columns representing categories of information and rows containing the actual data entries. The formats of the data extracts are described in the following tables.

Trading partner profile data extract

This record defines the trading partner profile settings being used to extract data. The fields are described in Table 93.

Table 93. Trading partner profile data extract fields

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **TPI**
Trading Partner Nickname	4–19	16	Char	Trading partner ID in WebSphere Data Interchange
Company Name	20–59	40	Char	Company name of the trading partner
Address line 1	60–99	40	Char	First line of the company's address
Address line 2	100–139	40	Char	Second line of the company's address
Comment line 1	140–179	40	Char	Can be used to further classify the trading partner (customer, supplier, division, subsidiary)
Comment line 2	180–219	40	Char	Can be used to further classify the trading partner (customer, supplier, division, subsidiary)

Table 93. Trading partner profile data extract fields (continued)

Label	Position	Length	Type	Description
Contact Name	220–249	30	Char	Name of a contact at this trading partner
Contact Phone	250–274	25	Char	Phone number of the contact
Network ID	275–282	8	Char	ID of the network used to communicate with this trading partner
Interchange ID	283–317	35	Char	Receiver ID used in interchanges to this trading partner
Account ID	318–349	32	Char	Network account ID
User ID	350–381	32	Char	Network user ID
Direction	382	1	Char	Direction of the (inbound or outbound)
Date of Last Transmission	383–390	8	Char	Interchange date of last transmission to this trading partner
Interchange control number	391–404	14	Char	Interchange control number of last transmission to this trading partner
Group control number	405–418	14	Char	Group control number of the last transmission to this trading partner
Transaction control number	419–432	14	Char	Transaction control number of the last transaction to this trading partner
Filler	433–1024	592	Char	Filler for expansion

Trading partner capability data extract

This record defines the trading partner capability settings being used to extract data. The fields are described in Table 94.

Table 94. Trading partner capability data extract fields

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **TPC**
Trading Partner Nickname	4–19	16	Char	Trading partner ID in WebSphere Data Interchange
Internal Trading Partner ID	20–54	35	Char	ID used for this trading partner internally (customer number, supplier number)
Company Name	55–94	40	Char	Company name of the trading partner
Address line 1	95–134	40	Char	First line of the company's address
Address line 2	135–174	40	Char	Second line of the company's address

Table 94. Trading partner capability data extract fields (continued)

Label	Position	Length	Type	Description
Comment line 1	175–214	40	Char	Can be used to further classify the trading partner (customer, supplier, division, subsidiary)
Comment line 2	215–254	40	Char	Can be used to further classify the trading partner (customer, supplier, division, subsidiary)
Direction	255	1	Char	Direction of the (inbound or outbound)
Standard ID	256–263	8	Char	ID of the EDI standard (X12, EDIFACT)
Version ID	264–265	2	Char	Version of the EDI standard
Release ID	266–267	2	Char	Release of the EDI standard
Description	268–317	50	Char	Description of the EDI standard/version/release
Transaction ID	318–325	8	Char	Transaction ID (such as ORDERS, DISPATCH, 850, 860)
Map ID	326–341	16	Char	Trading partner map name
Measurement date	342–349	8	Char	Date testing or production started with this map/trading partner combination
Measurement ID	350–353	4	Char	Type of statistic
Total number of transactions	354–368	15	Char	Total number of test or production transactions exchanged with this map/trading partner combination
Total errors	369–383	15	Char	Total number of test or production transactions exchanged with this map/trading partner combination that had errors
Appl Trading Partner	384–399	16	Char	Application trading partner ID
Filler	400–1024	625	Char	Filler for expansion

Network activity data extract

This record defines the network activity settings being used to extract data. The fields are described in Table 95.

Table 95. Network activity data extract fields

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **NTA**
Requestor ID	4–19	16	Char	Requestor ID
Network ID	20–27	8	Char	Network ID
Network name	28–57	30	Char	Network descriptive name

Table 95. Network activity data extract fields (continued)

Label	Position	Length	Type	Description
Account number	58–89	32	Char	Network account number
User ID	90–121	32	Char	Network user ID
Direction	122	1	Char	Direction of the transmission
Charge code	123	1	Char	Network charge code
Measurement ID	124–127	4	Char	Type of statistic
Day	128–135	8	Char	Measurement date
Interchange envelopes	136–150	15	Char	Total number of interchange envelopes
Total characters	151–165	15	Char	Total number of characters sent
Filler	166–1024	859	Char	Filler for expansion

Transaction activity data extract

This record defines the transaction activity settings being used to extract data. The fields are described in Table 96.

Table 96. Transaction activity data extract fields

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **TPA**
Trading Partner Nickname	4–19	16	Char	Trading partner ID in WebSphere Data Interchange
Internal Trading Partner ID	20–54	35	Char	ID used for this trading partner internally (such as customer number, supplier number)
Company Name	55–94	40	Char	Company name of the trading partner
Address line 1	95–134	40	Char	First line of the company's address
Address line 2	135–174	40	Char	Second line of the company's address
Comment line 1	175–214	40	Char	Can be used to further classify the trading partner (customer, supplier, division, subsidiary)
Comment line 2	215–254	40	Char	Can be used to further classify the trading partner
Direction	255	1	Char	Direction of the transmission
Standard ID	256–263	8	Char	ID of the EDI standard
Version ID	264–265	2	Char	Version of the EDI standard
Release ID	266–267	2	Char	Release of the EDI standard
Description	268–317	50	Char	Description of the EDI standard/version/release

Table 96. Transaction activity data extract fields (continued)

Label	Position	Length	Type	Description
Transaction ID	318–325	8	Char	Transaction ID
TPT ID	326–341	16	Char	Trading partner transaction ID
Data format ID	342–357	16	Char	Name of the data format
Measurement ID	358–361	4	Char	Type of statistic
Measurement Date	362–369	8	Char	Date of this statistic
Total transactions	370–384	15	Char	Total transactions for the indicated date
Total errors	385–399	15	Char	Total transactions in error for the indicated date
Appl Trading Partner	400–415	16	Char	Application trading partner ID
Filler	416–1024	609	Char	Filler for expansion

Document Store data extract information categories

You can extract information from the WebSphere Data Interchange Document Store databases using the ENVELOPE DATA EXTRACT, TRANSACTION DATA EXTRACT and DOCUMENT DATA EXTRACT commands. For information on DOCUMENT DATA EXTRACT command records, see “Document Store document data extract record formats” on page 321. See “Producing management reports from the Document Store” on page 9 for a description of these PERFORM commands. The categories of information that can be requested are described in Table 97.

Table 97. Document Store information categories

This value:	Requests:	Using this keyword:
E	Interchange data	INTERCHANGE(Y)
G	Group data	GROUP(Y)
T	Transaction data	TRANSACTION(Y)
A	Application data	APPLICATION(Y)
R	Transaction image	IMAGE(Y)
E, G, T	Send acknowledgment data	SENDACKDATA(Y)
E, G, T	Receive acknowledgment data	RECEIVEACKDATA(Y)
F, K	Send acknowledgment image	SENDACKIMAGE(Y)
F, K	Receive acknowledgment image	RECEIVEACKIMAGE(Y)

All data extracted from the Document Store is written to the EDIQUERY file. The information categories are either written as separate records by setting `CONCATENATE` to **N**, or combined and written as a single record by setting `CONCATENATE` to **Y**.

The following rules apply when requesting information categories:

1. Images (**R**, **F**, and **K**) are not produced unless you also request the transaction.
2. Images (**R**, **F**, and **K**) are always written as separate records even when concatenation has been requested.
3. Send acknowledgment data (**E**, **G**, and **T**) is not produced unless you also request the group record. If you want transaction acknowledgment data, you must also request the transaction record.
4. Send acknowledgment data (**E**, **G**, and **T**) is always concatenated to the corresponding group or transaction record even when concatenation has not been requested.
5. Receive acknowledgment data (**E**, **G**, and **T**) is not produced unless you also request the group or transaction record.
6. Receive acknowledgment data (**E**, **G**, and **T**) is always concatenated to the corresponding group or transaction record even when concatenation has not been requested.

Document Store data extract common key

All records created by Document Store Data Extract command have a common 141-byte key field that begins with a 3-character record ID. Portions of the key that do not apply to a particular record are initialized with blanks. Table 98 defines the common key format used to extract data, and shows which fields are used for which record types.

Table 98. Document Store common key format

Label	Position	Length	Type	Description
Record ID	1	3	Char	Record ID = **TX** Used for record types: A, E, G, K, R, T
Nickname	4	16	Char	Trading partner nickname Used for record types: A, E, G, K, R, T
Direction	20	1	Char	Direction of the transaction Used for record types: A, E, G, K, R, T
Control Number	21	14	Char	Interchange control number Used for record types: A, E, G, K, R, T
Receiver ID	35	35	Char	Interchange receiver ID Used for record types: A, E, G, K, R, T
Document ID	70	8	Char	Standard transaction ID
Control Number	78	14	Char	Group control number Used for record types: A, G, K, R, T
Control Number	92	14	Char	Transaction control number Used for record types: A, K, R, T
Controlling handle	106	20	Char	Handle value of controlling transaction (YYYYMMDDHHMMSSnnnnnn) Used for record types: A, K, R, T
Transaction handle	126	20	Char	Handle value of transaction (YYYYMMDDHHMMSSnnnnnn) Used for record types: A, K, R, T
Sequence Number	146	4	Char	Sequence number Used only for record type A

Document Store data extract record formats

The following tables show the formats for the records created by the TRANSACTION DATA EXTRACT command. All tables, except images, are padded to a length of 1024 bytes to leave room for expansion. If concatenation is requested in the WebSphere

Data Interchange Utility control statements, the full records as described are concatenated into a single record. Images are written separately and use the full logical record length of the EDIQUERY data set.

Interchange data extract record layout

This record defines the document settings being used to extract data. The fields are described in Table 99.

Table 99. Interchange data extract record layout

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **E**. **E1** = Functional acknowledgment **E2** = Transaction acknowledgment
Nickname	4–19	16	Char	Trading partner nickname.
Direction	20	1	Char	Direction of the interchange.
Control Number	21–34	14	Char	Interchange control number.
Receiver ID	35–69	35	Char	Interchange receiver ID.
Document ID	70-77	8	Char	Standard transaction ID
Filler	78–149	72	Char	Blanks.
Sender ID	150–184	35	Char	Interchange sender ID.
Fake flag	185	1	Char	No interchange header.
Sequence error flag	186	1	Char	Interchange out of sequence (ENVELOPE DATA EXTRACT only).
Usage indicator	187	1	Char	Type of usage.
Duplicate interchange flag	188	1	Char	Duplicate interchange received.
Envelope date	189–202	14	Char	Date/time envelope created (YYYYMMDDHHMMSS).
Send date	203–216	14	Char	Date/time envelope sent (YYYYMMDDHHMMSS).
TA1 acknowledgment	217	1	Char	TA1 acknowledgment received.
TA1 date	218–231	14	Char	Date/time TA1 received (YYYYMMDDHHMMSS).
Network status code	232–233	2	Char	Network status.
Network status text	234–253	20	Char	Network status code in text format.
Acknowledgment expected	254	1	Char	Network acknowledgment expected.
Acknowledgment received	255	1	Char	Network acknowledgment received.

Table 99. Interchange data extract record layout (continued)

Label	Position	Length	Type	Description
Acknowledged date	256–269	14	Char	Date/time envelope of network acknowledgment (YYYYMMDDHHMMSS).
Message user class	270–277	8	Char	Message user class assigned when sent.
Message name	278–285	8	Char	Message name assigned when sent.
Sequence number	286–290	5	Char	Sequence number assigned when sent.
Message ID	291–298	8	Char	Message ID assigned when sent.
Physical data set name	299–354	56	Char	Physical data set name to which data was queued.
Group count	355–363	11	Char	Number of groups in interchange.
Transaction count	364–376	11	Char	Number of transactions in interchange.
Segment count	377–387	11	Char	Number of segments in interchange.
Interchange size	388–398	11	Char	Number of bytes in interchange.
Interchange header	399–648	250	Char	Interchange header image.
Interchange trailer	649–678	30	Char	Interchange trailer image.
Filler	679–1024	354	Char	Reserved for expansion.

Group data extract record layout

This record defines the group settings being used to extract data. The fields are described in Table 100.

Table 100. Group data extract record layout

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **G**. **G1** = Functional acknowledgment **G2** = Transaction acknowledgment
Nickname	4–19	16	Char	Trading partner nickname.
Direction	20	1	Char	Direction of the group.
Control Number	21–34	14	Char	Interchange control number.
Receiver ID	35–69	35	Char	Interchange receiver ID.
Document ID	70-77	8	Char	Standard transaction ID
Control Number	78–91	14	Char	Group control number.
Filler	92–149	58	Char	Blanks.

Table 100. Group data extract record layout (continued)

Label	Position	Length	Type	Description
Fake flag	150	1	Char	Interchange did not have groups.
Sender ID	151–185	35	Char	Application sender ID.
Receiver ID	186–220	35	Char	Application receiver ID.
Acknowledgment expected	221	1	Char	Functional acknowledgment expected.
Acknowledgment received	222	1	Char	Functional acknowledgment received. **Note:** If an acknowledgment is not expected, this field is blank.
Acknowledgment date	223–236	14	Char	Date/time group acknowledgment received (YYYYMMDDHHMMSS).
Acknowledgment handle	237–256	20	Char	Handle value for the functional acknowledgment transaction (YYYYMMDDHHMMSSnnnnnn).
Transaction count	257–267	11	Char	Number of transactions in group.
Segment count	268–278	11	Char	Number of segments in group.
Group size	279–289	11	Char	Number of bytes in group.
Group header	290–442	153	Char	Image of the group header.
Group trailer	443–468	26	Char	Image of the group trailer.
Filler	469–1024	564	Char	Filler for expansion.

Transaction data extract record layout

This record defines the transaction settings being used to extract data. The fields are described in Table 101.

Table 101. Transaction data extract record layout

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **T.** **T1** = Functional acknowledgment **T2** = Transaction acknowledgment
Nickname	4–19	16	Char	Trading partner nickname.
Direction	20	1	Char	Direction.
Control Number	21–34	14	Char	Interchange control number.
Receiver ID	35–69	35	Char	Interchange receiver ID.
Document ID	70–77	8	Char	Standard transaction ID
Control Number	78–91	14	Char	Group control number.

Table 101. Transaction data extract record layout (continued)

Label	Position	Length	Type	Description
Control Number	92–105	14	Char	Transaction control number.
Controlling handle	106–125	20	Char	Handle value of the controlling transaction (YYYYMMDDHHMMSSnnnnnn).
Transaction handle	126–145	20	Char	Handle value of the transaction (YYYYMMDDHHMMSSnnnnnn).
Filler	146–149	4	Char	Blanks.
Enveloped date	150–163	14	Char	Date/time transaction was enveloped (YYYYMMDDHHMMSS).
Creation date	164–177	14	Char	Date/time transaction put into the store.
Transaction status code	178–179	2	Char	Current transaction status.
Transaction status text	180–199	20	Char	Transaction status in a text format.
Acknowledgment received	200	1	Char	Group acknowledgment received.
Acknowledgment received text	201–220	20	Char	Group acknowledgment received in a text format.
Acknowledgment date	221–234	14	Char	Date/time group acknowledgment was received (YYYYMMDDHHMMSS).
Trx Acknowledgment expected	235	1	Char	Transaction acknowledgment expected.
Trx Acknowledgment received	236	1	Char	Transaction acknowledgment received.
Trx Acknowledgment received text	237–256	20	Char	Transaction acknowledgment received in a text format.
Trx Acknowledgment date	257–270	14	Char	Date/time transaction acknowledgment was received (YYYYMMDDHHMMSS).
Acknowledgment handle	271–290	20	Char	Handle value of the transaction acknowledgment transaction (YYYYMMDDHHMMSSnnnnnn).
Segment count	291–301	11	Char	Number of segments in transaction.
Transaction size	302–312	11	Char	Number of bytes in transaction.
Enveloped segment count	313–323	11	Char	Number of segments in transaction when enveloped.
Enveloped transaction size	324–334	11	Char	Number of bytes in transaction when enveloped.
Format ID	335–350	16	Char	Last data format ID.
Transaction ID	351–358	8	Char	Standard transaction.

Table 101. Transaction data extract record layout (continued)

Label	Position	Length	Type	Description
Group ID	359–364	6	Char	Function Group ID associated with the transaction.
Envelope member	365–372	8	Char	Member name used for enveloping.
Envelope type	373	1	Char	Type of envelope associated with transaction.
Network ID	374–381	8	Char	Network associated with trading partner.
Standard ID	382–389	8	Char	EDI standard ID of the transaction.
Standard Version	390–391	2	Char	EDI standard version.
Standard Level	392–393	2	Char	EDI standard release level.
Internal Trading Partner ID	394–428	35	Char	Last internal trading partner ID.
Application control field	429–463	35	Char	Last application control field.
Delivery date	464–477	14	Char	Last date given/gotten from application (YYYYMMDDHHMMSS).
Earliest envelope date	478–485	8	Char	Earliest date that transaction will be enveloped (YYYYMMDD).
Earliest purge date	476–493	8	Char	Earliest date that transaction will be automatically purged (YYYYMMDD).
Error level	494	1	Char	Translation error level.
Store status	495–496	2	Char	Store status of the transaction.
Store status text	497–516	20	Char	Store status in a text format.
Override flag	517	1	Char	Envelope overrides provided for transaction.
Held flag	518	1	Char	Transaction held.
Test flag	519	1	Char	Test transaction.
Duplicate flag	520	1	Char	Transaction part of duplicate envelope.
Purge flag	521	1	Char	Purge transaction.
Translate flag	522	1	Char	Translated.
Detached flag	523	1	Char	Transaction detached.
Override handle	524–543	20	Char	Handle for enveloping overrides (YYYYMMDDHHMMSSnnnnnn).
Network security profile name	544–551	8	Char	Network security profile name for group level encryption.
Encryption key	552–567	16	Char	Encryption key name for group level encryption.
Authentication key	568–583	16	Char	Authentication key name for group level encryption.

Table 101. Transaction data extract record layout (continued)

Label	Position	Length	Type	Description
Application assigned control number	584–597	14	Char	Transaction control number if assigned by the application.
Data element delimiter	598	1	Char	Data element delimiter used when transaction was translated.
Sub element-delimiter	599	1	Char	Subelement delimiter used when transaction was translated.
Segment terminator	600	1	Char	Segment terminator used when transaction was translated.
Decimal notation	601	1	Char	Decimal notation used when transaction was translated.
Release character	602	1	Char	Release character used when transaction was translated.
Segment ID separator	603	1	Char	Segment ID separator used when transaction was translated.
Transaction header	604–688	85	Char	Image of the transaction header.
Transaction trailer	689–714	26	Char	Image of the transaction trailer.
Last ID	715–730	16	Char	Last used.
Filler	731–1032	302	Char	Reserved for expansion.

Application data extract record layout

This record defines the application settings being used to extract data. The fields are described in Table 102.

Table 102. Application data extract record layout

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID = **A**
Nickname	4–19	16	Char	Trading partner nickname
Direction	20	1	Char	Direction of format
Control Number	21–34	14	Char	Interchange control number
Receiver ID	35–69	35	Char	Interchange receiver ID
Document ID	70-77	8	Char	Standard transaction ID
Control Number	78–91	14	Char	Group control number
Control Number	92–105	14	Char	Transaction control number
Controlling handle	106–125	20	Char	Handle value of controlling transaction (YYYYMMDDHHMMSSnnnnnn)
Transaction handle	126–145	20	Char	Handle value of transaction (YYYYMMDDHHMMSSnnnnnn)

Table 102. Application data extract record layout (continued)

Label	Position	Length	Type	Description
Sequence Number	146–149	4	Char	Sequence number
Format ID	150–165	16	Char	Data format ID
Application ID	166–173	8	Char	Application ID
BATCH ID	174–181	8	Char	Batch ID
Delivery Date	182–195	14	Char	Date/time delivered to application (YYYYMMDDHHMMSS)
Error level	196	1	Char	Translation error level
Acceptable error level	197	1	Char	Acceptable error level
Error count	198–208	11	Char	Number of errors found
Filler	209–1024	824	Char	Reserved for expansion

Transaction/Acknowledgment image data extract record layout

This record defines the transaction and acknowledgment image settings being used to extract data. The fields are described in Table 103.

Table 103. Transaction/Acknowledgement image data extract record layout

Label	Position	Length	Type	Description
Record ID	1–3	3	Char	Record ID. Valid values are: **RX** = Transaction image continued **RZ** = Final transaction image record **FX** = Functional acknowledgment image continued **FZ** = Final functional acknowledgment image record **KX** = Transaction acknowledgment image continued **KZ** = Final transaction acknowledgment image record
Nickname	4–19	16	Char	Trading partner nickname.
Direction	20	1	Char	Direction of image data.
Control Number	21–34	14	Char	Interchange control number.
Receiver ID	35–69	35	Char	Interchange receiver ID.
Document ID	70-77	8	Char	Standard transaction ID
Control Number	78–91	14	Char	Group control number.
Control Number	92–105	14	Char	Transaction control number.
Controlling handle	106–125	20	Char	Handle value of controlling transaction (YYYYMMDDHHMMSSnnnnnn).
Transaction handle	126–145	20	Char	Handle value of transaction (YYYYMMDDHHMMSSnnnnnn).

Table 103. Transaction/Acknowledgement image data extract record layout (continued)

Label	Position	Length	Type	Description
Filler	146–149	4	Char	Blanks.
Total size	150–160	11	Char	Total size of the image.
Record size	168–171	11	Char	Size of image data in this record.
Image	172-end	Variable	Char	Transaction or acknowledgment image.

Document Store document data extract record formats

The following tables show the formats for the records created by the **DOCUMENT DATA EXTRACT** command. If concatenation is requested in the WebSphere Data Interchange Utility control statements, the full records as described are concatenated into a single record. Images are written separately and use the full logical record length of the EDIQUERY data set.

Document record (D) layout

This record defines the document settings being used to extract data. The fields are described in Table 104.

Table 104. Document record (D) layout

Label	Length	Position	Type	Description
Record ID	3	1–3	Char	Record ID = **D** or **DE** D = Document record DE = Document extended record
THANDLE	27	4–30	Char	Unique document identifier
SNDRID	35	31–65	Char	Sender id
SNDRQUAL	4	66–70	Char	Sender id qualifier
SNDRINTLID	35	71–105	Char	Sender internal id
SNDRNICKNAME	16	106–121	Char	Sender nickname / profile name
RCVRID	35	122–156	Char	Receiver id
RCVRQUAL	4	157–160	Char	Receiver id qualifier
RCVRINTLID	35	161–195	Char	Receiver internal id
RCVRNICKNAME	16	196–211	Char	Receiver nickname/profile id
DICTIONARY	30	212–241	Char	Dictionary
DOCUMENT	30	242–271	Char	Document
SYNTAX	3	272–274	Char	Syntax
DOCSUBTYPE	4	275–278	Char	
NETWORKID	8	279–286	Char	Network id over which the document was received or sent

Table 104. Document record (D) layout (continued)

Label	Length	Position	Type	Description
FLOWDIR	1	287	Char	Document flow direction (S/T)
ACFIELD1	127	288–414	Char	First 64 bytes of the application control field
PURGEINT	11	415–425	Char	Purge interval
PURGEDATE	27	426–452	Char	Purge date
DATECREATED	27	453–479	Char	Date created
GRPSNDRID	35	480–514	Char	Group sender id
GRPSNDRQUAL	4	515–518	Char	Group sender id qualifier
GRPRCVRID	35	519–553	Char	Group receiver id
GRPRCVRQUAL	4	554–557	Char	Group receiver id qualifier
BATCHID	8	558–565	Char	Batch identifier associated to this document
CHARENCODING	40	566–605	Char	Character set encoding for images
STORENCODING	4	606–609	Char	Image storage encoding option
DETACHED	11	610–620	Char	
DUPFLAG	11	621–631	Char	Flag used to indicate if this was a duplicate
HELDFLAG	11	632–642	Char	Flag used to indicate if the document was held
LASTMODIFIEDBY	27	643–669	Char	Document last modified by user
LASTSTATUS	8	670–677	Char	Status of the last activity performed on this document
OUTOFSEQ	11	678–688	Char	Flag to indicate if this document was out of sequence
PURGEFLAG	11	689–699	Char	Flag to indicate if this document need to be purged from the store
STATUSCODE	8	700–707	Char	Status code of the document
STORESTATUS	8	708–715	Char	Store status of the document
TOBEENVBY	27	715–742	Char	Last date by which this TRX has to be enveloped
TOBEPURGEDBY	27	743–769	Char	Last date by which this document needs to be purged
TRANSLATED	11	770–780	Char	Indicates if this document was translated
FAEXPECT	11	781–791	Char	Functional Acknowledgement Expected
FARCVD	11	792–802	Char	Function Acknowledgement Received
FARCVDDATE	27	803–829	Char	Functional Acknowledgement Received Date

Table 104. Document record (D) layout (continued)

Label	Length	Position	Type	Description
TRACKEXPECT	11	830–840	Char	Transport Acknowledgement Expected
TRACKRCVD	11	841–851	Char	Transport Acknowledgement Received
TRACKRCVDDATE	11	852–862	Char	Transport Acknowledgement Received Date

Document record extended (DE) layout

When EXTEND(Y) is specified in the **DOCUMENT DATA EXTRACT** command, the fields in Table 105 are appended to the end of the fields defined in Table 104 on page 321.

Table 105. Document record extended (DE) layout

Label	Length	Position	Type	Description
ACFIELD1	127	1–127	Char	Application control field1
ACFIELD2	127	128–255	Char	Application control field2
ACFIELD3	127	256–382	Char	Application control field3
ACFIELD4	127	383–509	Char	Application control field4
ACFIELD5	127	510–636	Char	Application control field5
FIELD1	127	637—763	Char	ICHGCTLNUM
FIELD2	127	764–809	Char	GRPCTLNUM
FIELD3	127	810–936	Char	TRXCTLNUM
FIELD4	127	937–1036	Char	REFCNT
FIELD5	127	1037–1190	Char	
FIELD6	127	1191–1317	Char	
FIELD7	127	1318–1444	Char	DOCID
FIELD8	127	1445–1571	Char	FAEXPECT
FIELD9	127	1572–1698	Char	FASTAT
FIELD10	127	1699–1825	Char	USGINDICATOR
FIELD11	127	1826–1952	Char	NWACKEXPECT
FIELD12	127	1953–2079	Char	NWACKSTAT
FIELD13	127	2080–2206	VARCHAR	SEQNO
FIELD14	127	2207–2333	VARCHAR	ENVTYPE
FIELD15	127	2334–2460	VARCHAR	OHANDLE
FIELD16	127	2461–2587	VARCHAR	MSGCLASS
FIELD17	127	2588–2714	VARCHAR	MSGHANDLE
FIELD18	127	2713–2841	VARCHAR	MSGSEQ

Table 105. Document record extended (DE) layout (continued)

Label	Length	Position	Type	Description
FIELD19	127	2842–2968	VARCHAR	MSGID
FIELD20	127	2969–3095	VARCHAR	TESTFLG

Image (I) record layout

Table 106. Image (I) record layout

Label	Length	Position	Type	Description
Record ID	2	0–1	Char	Record ID = **I** I = Image
	1	2	Char	X (more) or Z (last) this is the same as transaction images
THANDLE	27	3–29	Char	Unique document identifier
Total Size	11	30–40	Char	Sequential number
Record Size	11	41–51	Char	Row length
Image	Varies		Char	Type of image

Sample data:

```
I Z2006-11-14-20.19.22.859429 00000018468000000018468H01049591852TL          031229701223
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM® Director of Licensing
IBM Corporation
North Castle Drive
Armonk, N.Y. 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department DD40
P.O. Box 30021
Tampa, Florida 33630-3021
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the

names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

IBM
the IBM logo
AIX
CICS
IBMLink

IMS
WebSphere MQ
MVS
OS/390
WebSphere
z/OS

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Glossary of terms and abbreviations

This glossary defines WebSphere Data Interchange terms and abbreviations used in this book. If you do not find the term you are looking for, see the index or the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

This glossary includes terms and definitions from the *American National Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute. Copies may be ordered from the American National Standards Institute, 11 West 42 Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

A

AAR. Association of American Railroads. Represents the railroad industry in areas such as standards, public relations, and advertising.

acknowledgment. See *functional acknowledgment*, *network acknowledgment*.

ADF. *See data format*.

ANSI. American National Standards Institute.

ANSI ASC X12. ANSI Accredited Standards Committee X12, which develops and maintains generic standards for business transactions for EDI.

application. A program that processes business information. An application that requests services from WebSphere Data Interchange is an enabled application.

application data. The actual data in an application data file.

application data format. See *data format*.

application default profile. Identifies business applications, such as purchasing and accounts receivable, to WebSphere Data Interchange and sets specific WebSphere Data Interchange processing defaults for an application.

B

base structure. The data structure that contains all the data structures and data fields that define the application data for a single transaction.

binary format (BIN). Representation of a decimal value in which each field must be 2 or 4 bytes long. The sign (+ or -) is in the far left bit of the field, and the number value is in the remaining bits of the field. Positive numbers have a 0 in the sign bit. Negative numbers have a 1 in the sign bit and are in twos complement form.

C

CICS®. Customer Information Control System.

CD-ROM. Compact Disk-Read Only Memory; a storage medium for large amounts of data needed external to the personal computer.

client-server. A computing environment in which two or more machines work together to achieve a common task.

code list. A table, supplied by WebSphere Data Interchange or defined by the user, that contains all acceptable values for a single data field.

composite data element. In EDI standards, a group of related subelements, such as the elements that make up a name and address.

compound element. An item in the source or target document that contains child items. Examples are EDI segments and composite data elements, data format records and structures, and XML elements.

control number. Numbers (or masks used to create numbers) that are used to identify an interchange, group, or EDI transaction.

control string. An object compiled from a map, data format, and EDI standard transaction; it contains the instructions used by the translator to translate a document from one format to another.

control structure. The beginning and ending segments (header and trailer) of standard enveloped transmissions.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs.

customize. To alter to suit the needs of a company, such as removing from an EDI standard the segments and data elements that the company does not use.

D

data dictionary. A file containing the definitions of all the data elements of an EDI standard.

data element. A single item of data in an EDI standard, such as a purchase order number. Corresponds to a data field in a data format.

data element delimiter. A character, such as an asterisk (*), that follows the segment identifier and separates each data element in a segment. See also *element separator* and *segment ID separator*.

data field. A single item of data in a data format, such as a purchase order number. Corresponds to a data element in an EDI standard.

data format. A description of the application data for a particular transaction. A data format is composed of loops, records, data structures, and fields.

data format dictionary. A file that contains data format components.

data format record. A group of logically related fields set up as a record in a data format.

data format structure. A group of related data fields in a data format, such as the fields making up the line item of an invoice. Corresponds to a composite data element in an EDI standard.

DataInterchange/MVS™. The IBM DataInterchange product used on the host; pieces include a TSO parameter entry mechanism and a translator. The functionality available in this product is now available in WebSphere Data Interchange for z/OS.

DataInterchange/MVS-CICS. The CICS-based IBM DataInterchange product. The functionality available in this product is now available in WebSphere Data Interchange for z/OS.

data structure. A group of related data fields in a data format, such as the fields making up the line item of an invoice. Corresponds to a segment in a standard.

data transformation map. One of three supported map types. A data transformation map is a set of mapping instructions that describes how to translate data from a source document into a target document. Both the source and target documents can be one of several support document types.

DB2. Database 2, an IBM relational database management system.

ddname. Data definition name.

decimal notation. The character that represents a decimal point in the data.

delimiter. A character that terminates a string of characters, such as the value contained in a data element.

DI Client. WebSphere Data Interchange Client; the Windows-based, client/server interface for WebSphere Data Interchange.

dictionary. See *data dictionary*.

document. A business document that is exchanged between two enterprises as part of a business process, such as a purchase order or invoice. A document within WebSphere Data Interchange is singular. For example, it cannot contain multiple purchase orders. A document can also be represented in any syntax. For example, an XML purchase order and an EDI purchase order are both documents.

Document Type Definition (DTD). A list of all components included in the XML document and

their relationship to each other. This defines the structure of an XML document.

domain. The data structure or group of data structures in a data format to and from which you should restrict the mapping of EDI repeating segments and loops.

DTD. See *Document Type Definition*.

E

EDI. Electronic data interchange.

EDIA. Electronic Data Interchange Association.

EDI administrator. The person responsible for setting up and maintaining WebSphere Data Interchange.

EDI message. See message.

EDI standard. The industry-supplied, national, or international formats to which information is converted, allowing different computer systems and applications to interchange information.

EDI transaction. A single business document, such as an invoice.

EDI transaction set. A group of logically related data that make up an electronic business document, such as an invoice or purchase order.

EDIFACT. Electronic Data Interchange for Administration Commerce and Transport. See UN/EDIFACT.

electronic data interchange (EDI). A method of transmitting business information over a network, between business associates who agree to follow approved national or industry standards in translating and exchanging information.

electronic transmission. The means by which information is transferred between parties, such as over a public network.

element. See *data element*.

element separator. A character that separates the data elements in a segment. See also *data element delimiter*.

encryption. The encoding and scrambling of data. Data is encrypted by the sender and decrypted by the receiver using a predetermined program and unique electronic key.

event. An occurrence that is important to a user's computer tasks, such as a software error, sending a transaction, or acknowledging a message.

Extensible Markup Language (XML). A standard metalanguage for defining markup languages that was derived from, and is a subset of SGML. It is used to represent structured documents and data.

F

field. See *data field*.

floating segment. A segment of an EDI standard that may exist in many positions relative to other segments.

forward translation table. A user-defined table that translates data values that differ between trading partners. For example, if a manufacturer and supplier have different part numbers for the same item, each company can use its own part number and have it converted to the other company's part number during translation. Forward translation tables translate local values to standard values.

functional acknowledgment. An electronic acknowledgment returned to the sender to indicate acceptance or rejection of EDI transactions.

functional group. One or more transaction sets of a similar type transmitted from the same location, enclosed by functional group header and trailer segments.

G

global variable.. A variable that is shared among all instances of all documents within a translation session.

H

header. A control structure that indicates the start of an electronic transmission.

hierarchical loop. A technique for describing the relationship of data entities which are related in a parent/child manner, like a corporate organization chart. Used in mapping to group related data elements and segments such as trading partner address.

HL. *See hierarchical loop.*

I

IBM Global Network. The IBM communications network that provides products and services to IBM customers.

ICS. International Control Segments.

import. The process of taking WebSphere Data Interchange objects exported on another WebSphere Data Interchange system and incorporating them into the receiving system.

Information Exchange. A commerce engine of IBM Interchange Services for e-business that permits users to send and receive information electronically.

interchange. The exchange of information between trading partners.

J

JCL. Job Control Language.

K

key. In a profile member, the field that identifies the member. For example, the key for members of the trading partner profile is the trading partner nickname.

L

literal. In mapping, a value that is constant for each occurrence of the translation. If you provide the literal value during mapping, the translator does not have to refer repeatedly to the source to obtain the value.

local variable. A variable that is specific to the instance of the document in which it is being used.

log file. A file in which events are recorded.

logging. The recording of events in time sequence.

loop. A repeating group of related segments in a transaction set or a repeating group of related records and loops in a data format.

loop ID. A unique code identifying a loop and the number of times the group can be repeated.

loop repeat. A number indicating the maximum number of times a loop can be used in a transaction set.

M

mailbox. If you use a mail type protocol to exchange messages with your trading partners, you will have one or more registered mailboxes. The mailbox profile is used in WebSphere Data Interchange to define your mailboxes and any associated preferences.

map. A set of instructions that indicate to WebSphere Data Interchange how to translate data from one format to another.

map rule. An association between a data transformation map and a trading partner.

maximum use. A number indicating the maximum number of times a segment can be used in a transaction set or the maximum number of times that a data format loop or record can repeat.

message. A free-form, usually short, communication to a trading partner. In UN/EDIFACT standards, a group of logically related data that make up an electronic business document, such as an invoice. A message is equivalent to a document.

message log. The file in which WebSphere Data Interchange Client logs messages about errors that occur within the client.

multiple-occurrence mapping. A form of mapping in which all occurrences of a loop or repeating segment are mapped to the same repeating structure in the data format.

N

network acknowledgment. A response from the network indicating the status of an interchange envelope, such as sent or received.

network commands. The commands that you want WebSphere Data Interchange to pass to your network, defined in the network commands profile. In the host product, this file is named NETOP.

O

ODETTE. Organization for Data Exchange through Teletransmission in Europe.

P

parse. To break down into component parts.

path qualified mapping. A form of mapping in which all occurrences of a repeating compound or simple data element are mapped to a repeating compound or simple data element in another document.

PDS. Partitioned data set.

PDS members. Groups of related information stored in partitioned data sets.

profile. Descriptive information about trading partners, network connections, and so on. Each profile can contain one or more objects or members. For example, the trading partner profile contains members for your trading partners (one member for trading partner address).

program directory. A document shipped with each release of a product that describes the detailed content of the product.

Q

qualifier. A data element which gives a generic segment or data element a specific meaning. Qualifiers are used in mapping single or multiple occurrences.

R

receive map. One of three supported map types. A receive map is a set of mapping instructions that describe how to translate an EDI standard transaction into a proprietary application data document.

receive usage. An association between a receive map and a trading partner.

record. A logical grouping of related data structures and fields.

release character. The character that indicates that a separator or delimiter is to be used as text data instead of as a separator or delimiter. The release character must immediately precede the delimiter.

repository data. A group of data definitions, formats, and rules/usages, that WebSphere Data Interchange uses to process your data.

requestor. See *mailbox*.

reverse translation table. A user-defined table that translates data values that differ between trading partners. For example, if a manufacturer and supplier have different part numbers for the

same item, each company can use its own part number and have it converted to the other company's part number during translation. Reverse translation tables translate standard values to local values.

rule. See *map rule*.

runtime data. Data used by the WebSphere Data Interchange translator, such as control strings, code lists, translation tables and profiles.

S

security administrator. The person who controls access to business data and program functions.

segment. A group of related data elements. A segment is a single line in a transaction set, beginning with a function identifier and ending with a segment terminator delimiter. The data elements in the segment are separated by data element delimiters.

segment directory. A file containing the format of all segments in an EDI standard.

segment identifier. A unique identifier at the beginning of each segment consisting of two or three alphanumeric characters.

segment ID separator. The character that separates the segment identifier from the data elements in the segment.

segment terminator. The character that marks the end of a segment.

send map. On of three supported map types. A send map is a set of mapping instructions that describe how to translate a proprietary application data document into an EDI standard transaction.

send usage. An association between a send map and a trading partner.

simple element. An item in the source or target document that does not contain child items, only data. Examples are EDI data elements, data format fields, XML attributes, and PCDATA values.

single-occurrence mapping. A form of mapping in which each occurrence of a loop or repeating compound or simple data element in a document is mapped to a different compound or simple data element in another document.

source document definition. A description of the document layout that will be used to identify the format of the input document for a translation.

special literal. The send and receive Mapping Data Element Editors include the Literal or Mapping Command field. Literals are constant values you enter in this field, such as 123. Special literals are values you enter in this field that begin with an ampersand (&) and are command to WebSphere Data Interchange, rather than constant values. For example, to use today's date, you enter &DATE.

standards. See *EDI standard*.

structure. See *data structure* or *data format structure*.

subelement. In UN/EDIFACT standards, a data element that is part of a composite data element. For example, a data element and its qualifier are subelements of a composite data element.

subelement separator. A character that separates the subelements in a composite data element.

T

tag. In UN/EDIFACT standards, the segment identifier. In export/import, a code identifies each field in the export record. Such export/import files are known as "tagged" files.

target document definition. A description of the document layout that will be used to create an output document from a translation.

TD queue. See *transient data queue*.

TDCC. Transportation Data Coordinating Committee.

TDQ. Transient data queue.

temporary storage queue (TS). Storage locations reserved for immediate results in CICS. They are deleted after the task that created them is complete and they are no longer necessary.

TPT. Trading partner transaction. See *map*.

trading partner profile. The profile that defines your trading partners, including information about network account numbers, user IDs, who pays for network charges, etc.

trading partners. Business associates, such as a manufacturer and a supplier, who agree to exchange information using electronic data interchange.

trading partner transaction. See *map*.

trailer. A control structure that indicates the end of an electronic transmission.

transaction. A single business document, such as an invoice. See also *EDI transaction*.

transaction set. A group of standard data segments, in a predefined sequence, needed to provide all of the data required to define a complete transaction, such as an invoice or purchase order. See also *EDI transaction set*.

Transaction Store. The file that contains the results of translations and a history of translation activity.

transform. The process of converting a document from one format to another.

transient data queue (TD). A sequential data set used by the Folder Application Facility in CICS to log system messages.

translation. The process of converting a document from one format to another.

translation table. A user-defined table that translates data values that differ between trading partners. For example, if a manufacturer and supplier have different part numbers for the same item, each company can use its own part number and have it converted to the other company's part number during translation.

TSQ. See *temporary storage queue*.

U

UCS. Uniform Communication Standard.

unary operator. An operator that changes the sign of a numeric value.

UN/EDIFACT. United Nations Electronic Data Interchange for Administration Commerce and Transport.

Uniform Communication Standard (UCS). The EDI standard used in the grocery industry.

UN/TDI. United Nations Trade Data Interchange.

Usage. An association between a send or receive map and a trading partner.

V

validation table. A table, supplied by WebSphere Data Interchange or defined by the user, which contains all acceptable values for a single data field.

variable. The entity in which a value may be stored based on data received; as opposed to a constant value.

W

WebSphere Data Interchange. A generic term for the WebSphere Data Interchange products, WebSphere Data Interchange for z/OS and WebSphere Data Interchange for Multiplatforms. WebSphere Data Interchange is a translator of data from one document format to another; the pieces of this product include a TSO parameter entry mechanism, a CICS parameter entry mechanism, a Windows-based parameter entry mechanism (WebSphere Data Interchange Client), and a translator.

WebSphere Data Interchange Client. A Windows-based product for entry of parameters needed by the WebSphere Data Interchange translator.

WebSphere MQ. An IBM product that is used to implement messaging and queueing of data groups. Earlier releases of this product were known as MQSeries.

WebSphere MQ queue profile. Represents a relationship between a logical name and a physical WebSphere MQ queue name.

WINS. Warehouse Information Network Standard.

Windows®. Microsoft's graphical operating system under which WebSphere Data Interchange Client runs.

X

X12. A common EDI standard approved by the American National Standards Institute.

XML. See *Extensible Markup Language*.

Bibliography

This section describes the documentation available for the WebSphere Data Interchange product.

WebSphere Data Interchange publications

The WebSphere Data Interchange V3.3 publications are:

- *WebSphere Data Interchange for MultiPlatforms Quick Start Guide* CF0YREN
- *WebSphere Data Interchange for MultiPlatforms Administration and Security Guide* SC34-6214-01
- *WebSphere Data Interchange for MultiPlatforms Messages and Codes* SC34-6215-01
- *WebSphere Data Interchange for MultiPlatforms User's Guide* SC34-6216-01
- *WebSphere Data Interchange for MultiPlatforms Programmer's Reference* SC34-6217-01
- *WebSphere Data Interchange for MultiPlatforms Mapping Guide* SC23-5874-00
- *WebSphere Data Interchange for MultiPlatforms Utility Commands and File Formats Reference* SC23-5873-00
- *WebSphere Data Interchange for z/OS V3.3 Program Directory* GI10-2561-01
- *WebSphere Data Interchange for z/OS V3.3 Installation Guide* SC34-6269-01
- *WebSphere Data Interchange for z/OS V3.3 License File* GC34-6270-02

Softcopy books

All the WebSphere Data Interchange books are available in softcopy format.

Portable Document Format (PDF)

The library is supplied as stand-alone PDFs in US English in the DOC directory on the product CD. The contents of the DOC directory can be viewed without installing the product.

PDF files can be viewed and printed using the Adobe Acrobat Reader. You will need Adobe Acrobat Reader with Search Version 4.05 on Windows NT, or Adobe Acrobat Reader with Search Version 4.5 on UNIX® systems.

If you need to obtain the Adobe Acrobat Reader, or would like up-to-date information about the platforms on which the Acrobat Reader is supported, visit the Adobe Systems Inc. Web site at:

```
http://www.adobe.com/
```

If you cut and paste examples of commands from PDF files to a command line for execution, you must check that the content is correct before you press `Enter`. Some characters might be corrupted by local system and font settings.

WebSphere Data Interchange information available on the Internet

The WebSphere Data Interchange product Web site is at:

```
http://www.ibm.com/websphere/datainterchange/
```

By following links from this Web site you can:

- Obtain latest information about the WebSphere Data Interchange products.
- Access the WebSphere Data Interchange books in PDF format.

Bibliography

Index

Special characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Printed in USA

SC23-5873-00